



# Wrocław University of Science and Technology

---

Paweł Majewski MSc, Eng

**Machine learning methods for phenotyping insect biosystems on the  
example of the honeybee and the mealworm**

## **Doctoral Dissertation**

submitted to

**Wrocław University of Science and Technology**

**Field of Science:** Engineering and Technology

**Discipline of Science:** Information and Communication Technology

### **Supervisors**

Full Professor Robert Burduk, DSc, PhD, Eng  
Faculty of Information and Communication Technology  
Wrocław University of Science and Technology

Associate Professor Jacek Reiner, DSc, PhD, Eng  
Faculty of Mechanical Engineering  
Wrocław University of Science and Technology

**Keywords:** computer vision, semi-supervised learning, pseudo-labeling, object detection, segmentation, domain adaptation, re-identification, knowledge transfer, dense scenes

Wrocław, September 2024



*The best moments in our lives, are not the passive, receptive, relaxing times (...) The best moments usually occur when a person's body or mind is stretched to its limits in a voluntary effort to accomplish something difficult and worthwhile*

– Mihály Csíkszentmihályi





# Acknowledgements

---

Pragnę wyrazić moją głęboką wdzięczność moim promotorom, profesorom Robertowi Burdukowi i Jackowi Reinerowi, za ich wsparcie podczas realizacji mojej pracy doktorskiej. Dziękuję za zaufanie, cierpliwość, życzliwość oraz nieocenioną pomoc przy realizacji dodatkowych przedsięwzięć, takich jak konferencje międzynarodowe i staż zagraniczny. Szczególne podziękowania kieruję do profesora Jacka Reintera, którego wsparcie, motywacja i inspiracja od wczesnych etapów mojej kariery naukowej, począwszy od pracy inżynierskiej i magisterskiej, były kluczowe w podjęciu pracy doktorskiej. Jestem za to bardzo wdzięczny.

Wyjątkowe podziękowania składam moim rodzicom, Marioli i Henrykowi Majewskim, za ich nieustanne wsparcie na każdym etapie mojej edukacji i kariery naukowej. Dziękuję za pomoc w chwilach zwątpienia oraz za stworzenie domu w Łomnicy, do którego zawsze chętnie wracam. Mojej siostrze Asi oraz bratu Tomkowi dziękuję za wzajemne wsparcie a całej rodzinie za wiarę we mnie.

Dziękuję również moim kolegom z zespołu MVLab za fantastyczną atmosferę i wspólną realizację projektów. W szczególności dziękuję Piotrowi Lampie za pomoc w opracowywaniu systemów wizyjnych, które były podstawą moich badań od początku doktoratu.

*Wrocław, September 2024*

*Paweł*



# Abstract

---

Recently, we have observed a significant increase in the importance of machine learning (ML) and computer vision (CV) methods in more areas of fundamental research and application problems. Considering sustainable development and human well-being, agriculture is one of the essential fields for applying ML/CV methods. The analysis of insect biosystems, thematically associated with agriculture, is an important research area for ML/CV methods both in terms of research gaps and high application potential. The task of ML/CV methods in the context of biosystems is to phenotype them, i.e. to calculate highly informative indicators that characterize a given biosystem. This dissertation focuses on the honeybee and mealworm biosystems.

There are numerous papers in the literature solving successive application problems for precision insect farming. However, we still can find research gaps at the level of developing efficient and robust ML methods. The problems of weakly represented datasets [RG1] and dense scenes [RG2] are common in developing ML methods for phenotyping insect biosystems and involve the difficulty of obtaining a representative dataset with reasonable time spent on labelling. In most papers, researchers focused only on training and evaluating ML models under the assumption of having a representative dataset, omitting the critical step of efficiently developing a representative dataset. The articles also did not consider methods for supervising the performance of models and their adaptation during production [RG3], which, in the context of the occurring changeability of biosystems over time, is a significant issue. It should also be noted that a considerable number of solutions in the area of phenotyping insect biosystems are based on off-the-shelf models, so still a reasonable area of research is dedicated methods to the problems of phenotyping insect biosystems, including the issue of taking into account domain knowledge [RG4]. Near-real-time inference requirements for the problems under consideration favour low-complexity solutions. Methods for reducing complexity and inference time [RG5] are another important research issue. Most work in the literature is based on phenotyping insect biosystems at the population level without considering the characteristics of individuals. Phenotyping at the individual level [RG6] represents another research gap.

At the same time, universal machine learning methods can also be found in the literature, which can be useful in developing dedicated solutions for phenotyping insect biosystems. Generated synthetic images, which are a special type of augmentation, make it possible to reduce the time spent on annotation. Semi-supervised learning allows unlabeled samples to be included in model training or adaptation, increasing the final model efficiency. Knowledge transfer techniques provide a basis for training a new model based on the prediction of another model or method, reducing the final complexity of the solution. End-to-end architectures provide condensed solutions under specific application problems. Significant advances in

---

re-identification are seen for more types of objects, including animals.

Taking into account the research gaps and state-of-the-art discussed, the following research hypothesis was formulated: 'Machine learning methods using synthetic images, semi-supervised learning, knowledge transfer and end-to-end architectures enable the development of dedicated models for phenotyping insect biosystems that are more efficient, easier to develop and maintain and characterized by shorter inference times than currently used machine learning methods' and research objectives: [O1] development of method enabling faster development of ML methods for phenotyping insect biosystems, involving synthetic image generation and semi-supervised learning (pseudo-labeling), [O2] development of method enabling more efficient maintenance during the production of ML methods for phenotyping insect biosystems, involving detecting domain shift (or concept drift) effect and adaptation technique, [O3] development of method enabling reduction of complexity (inference time) of ML methods for phenotyping insect biosystems, involving knowledge transfer and end-to-end model, [O4] development of method enabling the incorporation of domain knowledge (a priori) in the development, maintenance, and inference of ML methods for phenotyping insect biosystems, and [O5] development of method enabling phenotyping insect biosystems at the level of individuals (rather than population), involving re-identification and detection of behavioural patterns.

The doctoral dissertation is in the form of a collection of six thematically related scientific articles published in scientific journals or in peer-reviewed proceedings of international conferences, and one article that is currently under review. The articles included in the dissertation address the common problem of phenotyping insect biosystems.

The article [A1] (*Multipurpose monitoring system for edible insect breeding based on machine learning*) proposed a 3-module system for monitoring the rearing of the mealworm. The first module was based on the Mask-CNN model and was used for instance segmentation of the growth stages of the mealworm (live larva, pupa, beetle) and anomalies (dead larva, pest). The second module was based on the U-Net model and was related to the semantic segmentation of chitinous moults and feed. The third module was responsible for calculating size indices of larvae (length, volume) at the level of individuals and the entire population. Synthetic images with automatically generated labels were used to train the ML models, significantly reducing the labelling time [O1] of images representing dense scenes [RG2].

In the article [A2] (*Prediction of the remaining time of the foraging activity of honey bees using spatio-temporal correction and periodic model re-fitting*), a model was developed to predict the remaining time of the daily foraging activity of bees based on the current and past activity at the entrance of the hive (understood as the number of registered bees on consecutive frames), the time until sunset and environmental factors (temperature, humidity). To maintain the high accuracy of the prediction model [RG3], a method of periodic re-fitting of the model based on automatically generated target values was proposed [O2]. In determining the target values, domain knowledge [RG4] was taken into account through the spatio-temporal correction method [O4], which significantly reduced the error progression during periodic model re-fitting. The article confirms the possibility of maintaining high accuracy of the prediction model, when concept drift occurs, throughout the beekeeping season.

The article [A3] (*Monitoring the growth of insect larvae using a regression convolutional neural*

---

*network and knowledge transfer*) focused on developing a method for phenotyping larvae with reduced complexity and inference time [RG5], compared to the method proposed in [A1]. The developed solution was a multioutput regression convolutional neural network trained using knowledge transfer [O3]. To train the model, the size indices of the larvae obtained in the multistage phenotyping procedure using classical CV methods and the larvae segmentation model (trained on synthetic images) were used while automating the labelling process [O1] of images representing dense scenes [RG2]. For calibration purposes, only a few labelled samples were used.

The article [A4] (*Mixing Augmentation and Knowledge-Based Techniques in Unsupervised Domain Adaptation for Segmentation of Edible Insect States*) addressed the task of domain adaptation [RG3] for insect biosystem phenotyping problems using the example of segmentation of selected states of the mealworm (live larva, dead larva, pupa). A 2-stage domain adaptation method was proposed [O2], where after each stage model training was carried out on a new set of prepared samples. The first stage of the developed method was based on generating synthetic images using a pool of objects from the source domain extended with augmented objects. In the second stage, a pool of objects from the target domain was proposed using the model prediction from stage one. The objects from the target pool were filtered based on domain knowledge [RG4] and then used to generate synthetic images [O4].

The article [A5] (*Improved Pest Detection in Insect Larvae Rearing with Pseudo-Labeling and Spatio-Temporal Masking*) focused on the problem of weakly represented datasets [RG1] in the context of pest detection. The paper proposed a method for developing a pest detection model assuming a small initial set of labelled samples [O1]. The developed method was based on generating pseudo-labels based on the prediction of a previously trained model. A spatio-temporal masking method [O4] based on domain knowledge [RG4] was responsible for reducing errors in pseudo-label generation. The proposed solution also included identifying positive samples (images with pests) for further labelling from many samples acquired daily.

The article [A6] (*End-to-end Solution for Tenebrio Molitor Rearing Monitoring with Uncertainty Estimation and Domain Shift Detection*) focused on the development of a condensed end-to-end architecture [RG5] for phenotyping the mealworm biosystem incorporating the functionalities of the separate modules proposed in [A1]. The proposed solution extended the YOLOv8 architecture with additional heads (branches) related to the corresponding tasks (estimation of image coverage coefficients of feed and chitinous moults, phenotyping of larvae) [O3]. The paper also proposed a method for estimating prediction uncertainty with the detection of the domain shift phenomenon [RG3], using model ensemble and bootstrapping [O2]. Training of successive branches of the model was performed using separate datasets prepared for a specific problem [O1], effectively reducing the time needed to label images representing dense scenes [RG2].

The article [A7] (*Phenotyping with dynamic characteristics determination for Tenebrio Molitor beetles in selective breeding using re-identification*) addressed the problem of phenotyping mealworm beetles at the level of individuals [RG6]. The paper proposed a procedure for developing beetle re-identification models that did not require manual labelling of samples based on obtaining samples of individuals during the training stage when the beetles were isolated from each other in stations [O1]. Using the re-identification model in the testing stage, the dynamic

---

characteristics of individuals were determined, including the detection of mating behaviour pattern [O5]. The paper also proposed a method for the initial selection of individuals for phenotyping based on the designed hybrid metric. A domain adaptation [RG3] method based on supplementing the training set with samples from the target domain with automatically determined pseudo-labels was developed to reduce the domain shift phenomenon occurring between the training and testing stages [O2].

In summary, the research carried out, as part of the dissertation, confirmed the formulated research hypothesis, i.e. machine learning methods using synthetic images, semi-supervised learning, knowledge transfer and end-to-end architectures enable the development of dedicated models for phenotyping insect biosystems that are more efficient, easier to develop and maintain and characterized by shorter inference times than currently used machine learning methods. Furthermore, the proposed techniques for the introduction of domain knowledge and re-identification strengthened the statement that dedicated methods could be developed for phenotyping insect biosystems. All set research objectives (O1-O5) were met and defined research gaps (RG1 - RG6) were filled.

# Streszczenie

---

W ostatnim czasie obserwujemy znaczący wzrost znaczenia metod uczenia maszynowego (UM) i widzenia komputerowego (WK) w coraz większej liczbie obszarów badań podstawowych i problemów aplikacyjnych. Biorąc pod uwagę zrównoważony rozwój i dobrostan ludzi, rolnictwo jest jednym z kluczowych obszarów zastosowania metod UM/WK. Analiza biosystemów owadów, tematycznie związana z rolnictwem, jest ważnym obszarem badawczym dla metod UM/WK zarówno pod względem luk badawczych, jak i wysokiego potencjału aplikacyjnego. Zadaniem metod UM/WK w kontekście biosystemów jest ich fenotypowanie, czyli obliczanie wysokoinformatywnych współczynników charakteryzujących dany biosystem. W pracy doktorskiej skupiono się na biosystemach pszczoły miodnej i mącznika młynarka.

W literaturze możemy odnaleźć znaczącą liczbę prac rozwiązujących kolejne problemy aplikacyjne dla precyzyjnej hodowli owadów. Jednakże na poziomie opracowywania efektywnych i odpornych metod UM nadal odnajdziemy luki badawcze. Problem słabo reprezentowanych zbiorów danych [RG1] oraz gęstych scen [RG2] jest często spotykany przy rozwijaniu metod UM dla fenotypowania biosystemów owadów i wiąże się z trudnością uzyskania reprezentatywnego zbioru danych przy racjonalnym czasie spędzonym na etykietowanie. W większości prac badacze skupiali się tylko na procesie treningu i ewaluacji modeli UM przy założeniu posiadania reprezentatywnego zbioru danych, omijając bardzo istotny etap efektywnego opracowywania reprezentatywnego zbioru danych. Artykuły również nie uwzględniały metod nadzorowania działania modeli i ich adaptacji w czasie produkcji [RG3], co w kontekście występującej zmienności biosystemów w czasie jest zagadnieniem znaczącym. Należy również zwrócić uwagę na to, że spora liczba rozwiązań z obszaru fenotypowania biosystemów owadów opiera się na gotowych modelach i nadal zasadnym obszarem badań są metody dedykowane problemom fenotypowania biosystemów owadów łącznie z zagadnieniem uwzględniania wiedzy dziedzinowej [RG4]. Wymagania odnośnie czasu wnioskowania bliskiego rzeczywistego dla rozważanych problemów faworyzują rozwiązania o małej złożoności. Metody redukcji złożoności i czasu wnioskowania [RG5] są kolejnym ważnym zagadnieniem wymagającym badań. W literaturze przeważająca liczba prac opiera się na fenotypowaniu biosystemów owadów na poziomie populacji bez uwzględniania indywidualnej charakterystyki osobników. Fenotypowanie na poziomie osobników [RG6] stanowi kolejną lukę badawczą.

Jednocześnie w literaturze odnajdziemy również uniwersalne metody uczenia maszynowego, które mogą okazać się pomocne przy rozwijaniu dedykowanych rozwiązań dla fenotypowania biosystemów owadów. Generowane obrazy syntetyczne, stanowiące specjalny rodzaj augmentacji, umożliwiają skrócenie czasu spędzonego na adnotację. Uczenie częściowo-nadzorowane pozwala uwzględnić próbki nieetykietowane w treningu lub adaptacji modelu,

---

zwiększając ostateczną efektywność modelu. Technika transferu wiedzy daje podstawę do treningu nowego modelu na bazie predykcji innego modelu lub metody, redukując ostateczną złożoność rozwiązania. Architektury end-to-end stanowią skondensowane rozwiązania pod konkretne problemy aplikacyjne. Znaczne postępy w re-identyfikacji są zauważalne dla kolejnych rodzajów obiektów, również zwierząt.

Biorąc pod uwagę omówione luki badawcze oraz stan wiedzy, sformułowano następującą hipotezę badawczą: 'Metody uczenia maszynowego wykorzystujące obrazy syntetyczne, uczenie częściowo-nadzorowane, transfer wiedzy oraz architektury end-to-end umożliwiają opracowywanie modeli dedykowanych dla fenotypowania biosystemów owadów, które są bardziej efektywne, łatwiejsze w rozwijaniu i utrzymaniu oraz charakteryzują się krótszym czasem wnioskowania w porównaniu do aktualnie wykorzystywanych metod uczenia maszynowego' oraz cele badań: [O1] opracowanie metody umożliwiającej szybsze rozwijanie metod UM do fenotypowania biosystemów owadów, włączając w to generowanie obrazów syntetycznych oraz uczenie częściowo nadzorowane, [O2] opracowanie metody umożliwiającej bardziej efektywne utrzymanie modeli UM w czasie produkcji dla fenotypowania biosystemów owadów, włączając w to metody detekcji efektu przesunięcia domeny (lub dryftu koncepcji) oraz metody adaptacji, [O3] opracowanie metody umożliwiającej redukcję złożoności (czasu wnioskowania) metod UM do fenotypowania biosystemów owadów, włączając w to transfer wiedzy oraz modele end-to-end, [O4] opracowanie metody umożliwiającej wprowadzanie wiedzy dziedzinowej (a priori) w proces opracowania, utrzymania oraz wnioskowania metod UM do fenotypowania biosystemów owadów oraz [O5] opracowanie metody umożliwiającej fenotypowanie biosystemów owadów na poziomie osobników (w przeciwieństwie do populacji), włączając w to re-identyfikację oraz wykrywanie wzorców zachowania.

Dysertacja doktorska jest w formie cyklu sześciu powiązanych tematycznie artykułów naukowych opublikowanych w czasopismach naukowych lub w recenzowanych materiałach z konferencji międzynarodowych oraz jednego artykułu, który aktualnie jest w trakcie recenzji. Zawarte artykuły w dysertacji obejmują wspólną problematykę fenotypowania biosystemów owadów.

Artykuł [A1] (*Multipurpose monitoring system for edible insect breeding based on machine learning*) zaproponował 3-modułowy system do monitoringu hodowli mącznika młynarka. Pierwszy moduł był oparty o model Mask-CNN i służył do segmentacji instancyjnej stadiów rozwojowych mącznika młynarka (larwa żywa, poczwarka, chrząszcz) oraz anomalii (larwa martwa, szkodnik). Drugi moduł był oparty o model U-Net i był związany z segmentacją semantyczną wyniki chitynowej oraz paszy. Trzeci moduł odpowiadał za obliczanie wskaźników wielkościowych larw (długość, objętość) na poziomie osobników oraz całej populacji. Do treningu modeli UM wykorzystano obrazy syntetyczne z automatycznie generowanymi etykietami, co znacznie zmniejszyło czas etykietowania [O1] obrazów reprezentujących gęste sceny [RG2].

W artykule [A2] (*Prediction of the remaining time of the foraging activity of honey bees using spatio-temporal correction and periodic model re-fitting*) opracowano model predykcji czasu pozostałego do końca dziennego oblotu pszczół na podstawie aktualnej i minionej aktywności na wejściu do ula (rozumianej jako ilość zarejestrowanych pszczół na kolejnych klatkach), czasu do zachodu słońca oraz parametrów środowiskowych (temperatura, wilgot-



---

ność). Dla utrzymania wysokiej dokładności modelu predykcji [RG3] zaproponowano metodę okresowego dopasowywania modelu na podstawie automatycznie generowanych wartości docelowych [O2]. Przy wyznaczaniu wartości docelowych uwzględniono wiedzę dziedzinową [RG4] poprzez metodę korekcji przestrzenno-czasowej [O4], co znacznie ograniczało progresję błędu podczas okresowego dopasowywania modelu. W artykule potwierdzono możliwość utrzymania wysokiej dokładności modelu predykcji, podczas występowania zjawiska dryftu koncepcji, przez cały sezon pszczelarski.

Artykuł [A3] (*Monitoring the growth of insect larvae using a regression convolutional neural network and knowledge transfer*) skupia się na opracowaniu metody fenotypowania larw o zmniejszonej złożoności i czasie wnioskowania [RG5], w porównaniu do metody zaproponowanej w [A1]. Opracowane rozwiązanie to regresyjna wielowyjściowa sieć konwolucyjna trenowana z wykorzystaniem transferu wiedzy [O3]. Do treningu modelu wykorzystano wskaźniki wielkościowe larw uzyskane w procesie wieloetapowego fenotypowania z wykorzystaniem klasycznych metod WK oraz modelu segmentacji larw (trenowanego na obrazach syntetycznych), jednocześnie automatyzując proces etykietowania [O1] obrazów reprezentujących gęste sceny [RG2]. Dla celów kalibracyjnych wykorzystano jedynie parę etykietowanych próbek.

Artykuł [A4] (*Mixing Augmentation and Knowledge-Based Techniques in Unsupervised Domain Adaptation for Segmentation of Edible Insect States*) podejmuje problem adaptacji domeny [RG3] dla problemów fenotypowania biosystemów owadów na przykładzie segmentacji wybranych stanów mącznika młynarka (larwa żywa, larwa martwa, poczwarka). Zaproponowano 2-etapową metodę adaptacji domeny [O2], gdzie po każdym z etapów przeprowadzano trening modelu na nowym zbiorze przygotowanych próbek. Pierwszy etap opracowanej metody bazował na generowaniu obrazów syntetycznych z wykorzystaniem puli obiektów z domeny źródłowej rozszerzonej o augmentowane obiekty. W drugim etapie, korzystając z predykcji modelu z etapu pierwszego, zaproponowano pulę obiektów pochodzących z domeny docelowej. Obiekty z puli docelowej poddano filtracji opartej na wiedzy dziedzinowej [RG4] a następnie wykorzystano do generowania obrazów syntetycznych [O4].

Artykuł [A5] (*Improved Pest Detection in Insect Larvae Rearing with Pseudo-Labeling and Spatio-Temporal Masking*) skupia się na problemie słabo reprezentowanych zbiorów danych [RG1] w kontekście detekcji szkodników. W artykule zaproponowano metodę rozwijania modelu detekcji szkodników przy założeniu małego początkowego zbioru etykietowanych próbek [O1]. Opracowana metoda bazowała na generowaniu pseudoetykiet na podstawie predykcji uprzednio wytrenowanego modelu. Za zmniejszenie błędów przy generowaniu pseudoetykiet odpowiadała metoda maskowania przestrzenno-czasowego [O4] bazująca na wiedzy dziedzinowej [RG4]. Zaproponowane rozwiązanie obejmowało również identyfikację pozytywnych próbek (obrazów z pasożytem) do dalszego etykietowania spośród dużej ilości próbek pozyskiwanych dziennie.

Artykuł [A6] (*End-to-end Solution for Tenebrio Molitor Rearing Monitoring with Uncertainty Estimation and Domain Shift Detection*) skupiał się na opracowaniu skondensowanej architektury end-to-end [RG5] do fenotypowania biosystemu mącznika młynarka, obejmującej funkcjonalności oddzielnych modułów zaproponowanych w [A1]. Zaproponowane rozwiązanie polegało na rozszerzeniu architektury YOLOv8 o dodatkowe głowy (gałęzie) związane

---

z odpowiednimi zadaniami (estymacja współczynników pokrycia obrazu paszą i wylinką chitynową, fenotypowanie larw) [O3]. W artykule zaproponowano również metodę estymacji niepewności predykcji wraz z detekcją zjawiska przesunięcia domeny [RG3], wykorzystując zespół modeli oraz bootstrapping [O2]. Trening kolejnych gałęzi modelu wykonywano z wykorzystaniem osobnych zbiorów danych przygotowanych pod konkretny problem [O1], skutecznie redukując czas potrzebny na etykietowanie obrazów reprezentujących gęste sceny [RG2].

Artykuł [A7] (*Phenotyping with dynamic characteristics determination for Tenebrio Molitor beetles in selective breeding using re-identification*) podejmował problem fenotypowania chrząszczy mącznika młynarka na poziomie pojedynczych osobników [RG6]. W pracy zaproponowano procedurę opracowania modeli re-identyfikacji chrząszczy niewymagającą ręcznego etykietowania próbek opartą o pozyskiwanie próbek osobników w fazie treningowej, gdy chrząszcze były odizolowane od siebie w stanowiskach [O1]. Wykorzystując model re-identyfikacji w fazie testowej wyznaczano dynamiczne charakterystyki osobników, w tym wykrywano wzorce zachowania w postaci krycia [O5]. W artykule zaproponowano również metodę wstępnego wyboru osobników do fenotypowania na podstawie opracowanej hybrydowej metryki. Dla redukcji zjawiska przesunięcia domeny [RG3], występującej pomiędzy etapem treningowym i testowym, opracowano metodę adaptacji domeny bazującą na uzupełnieniu zbioru treningowego o próbki z domeny docelowej z automatycznie wyznaczanymi pseudoetykietami [O2].

Podsumowując, przeprowadzone badania, w ramach w pracy doktorskiej, potwierdziły postawioną hipotezę badawczą, czyli metody uczenia maszynowego wykorzystujące obrazy syntetyczne, uczenie częściowo-nadzorowane, transfer wiedzy oraz architektury end-to-end umożliwiają opracowywanie modeli dedykowanych dla fenotypowania biosystemów owadów, które są bardziej efektywne, łatwiejsze w rozwijaniu i utrzymaniu oraz charakteryzują się krótszym czasem wnioskowania w porównaniu do aktualnie wykorzystywanych metod uczenia maszynowego. Ponadto zaproponowane techniki wprowadzania wiedzy dziedzinowej oraz re-identyfikacji utwierdziły stwierdzenie o możliwości opracowywania dedykowanych metod do fenotypowania biosystemów owadów. Wszystkie postawione cele badawcze (O1-O5) zostały zrealizowane a zdefiniowane luki badawcze (RG1 - RG6) uzupełnione.

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Related Work . . . . .	2
1.3	Research Gaps . . . . .	6
1.4	Research Hypothesis and Objectives . . . . .	15
<b>2</b>	<b>Results Summary</b>	<b>17</b>
2.1	Publications . . . . .	17
2.2	Other Important Achievements . . . . .	19
2.3	Achieving Research Objectives . . . . .	21
<b>3</b>	<b>Conclusion and Future Work</b>	<b>35</b>
<b>4</b>	<b>Publications</b>	<b>37</b>
4.1	Multipurpose monitoring system for edible insect breeding based on machine learning . . . . .	37
4.2	Prediction of the remaining time of the foraging activity of honey bees using spatio-temporal correction and periodic model re-fitting . . . . .	53
4.3	Monitoring the growth of insect larvae using a regression convolutional neural network and knowledge transfer . . . . .	64
4.4	Mixing Augmentation and Knowledge-Based Techniques in Unsupervised Domain Adaptation for Segmentation of Edible Insect States . . . . .	82
4.5	Improved Pest Detection in Insect Larvae Rearing with Pseudo-Labeling and Spatio-Temporal Masking . . . . .	91
4.6	End-to-end Solution for Tenebrio Molitor Rearing Monitoring with Uncertainty Estimation and Domain Shift Detection . . . . .	100
4.7	Phenotyping with dynamic characteristics determination for Tenebrio Molitor beetles in selective breeding using re-identification . . . . .	111
	<b>Bibliography</b>	<b>145</b>



# CHAPTER 1

## Introduction

---

### 1.1 Background

Recently, we have observed a significant increase in the importance of machine learning (ML) and computer vision (CV) methods in more areas of fundamental research and application problems. The sharp rise in ML and CV in recent years can be attributed to several key factors: (1) growth in computing power (GPU, TPU, cloud computing), (2) large datasets (Big Data), (3) progress in algorithms and models (deep learning[48], convolutional neural networks[43], transfer learning, transformers[97], generative models[32]), and (4) frameworks and open-source libraries (TensorFlow, PyTorch, Scikit-Learn, OpenCV). Among the most popular application areas for ML/CV methods can be mentioned: manufacturing[71], healthcare[28], finance[2], transportation[112], marketing[64], entertainment[12], and agriculture[90].

Considering sustainable development and human well-being, agriculture is one of the essential fields for applying ML/CV methods. The growing global human population is necessitating an increase in food production. Simultaneously, consumer demands for the quality of produced food are also increasing, especially in the context of residues of harmful chemicals (e.g. pesticides, antibiotics) in the final product. Many consumers prefer ecological products. Attention is also being paid to environmental costs in food production, i.e. greenhouse gas emissions and water consumption.

Precision agriculture (PA) and precision livestock farming (PLF) respond to the problems mentioned. Through the use of the latest data analysis methods, attempts are made to monitor and control the process of crop and livestock farming to achieve a high quantity and quality of product at the lowest possible environmental and financial cost. The nomenclature separates the terms 'precision agriculture' and 'precision livestock farming' [29] due to the different characteristics of the objects observed and dedicated methods for analysis. In the case of livestock, observing current behaviour and dynamic patterns plays a significant role, while in the case of plants, recording long-term changes is often sufficient. Precision insect farming (PIF), including precision beekeeping (PB), is another important area. Calling farmed insects (bees) as 'livestock' is not wrong, but the term is more likely to be associated with cattle, pigs and poultry in the literature. Undoubtedly, we can also find characteristic features of PIF at the level of data analysis methods. In contrast to 'livestock', the analyzed insect

population is most often characterized by tens of thousands of individuals, which tends to favour analyzing at the population level. However, there are exceptions to this rule, and individualized analysis of insects is reasonable in special cases such as selective breeding studies. The unifying term for PA, PLF, and PIF/PB can be biosystem phenotyping, which is based on the determination of highly informative features (phenotyping) that enable the description of the studied biosystems. Figure 1.1 shows the presented classification for biosystem phenotyping problems.

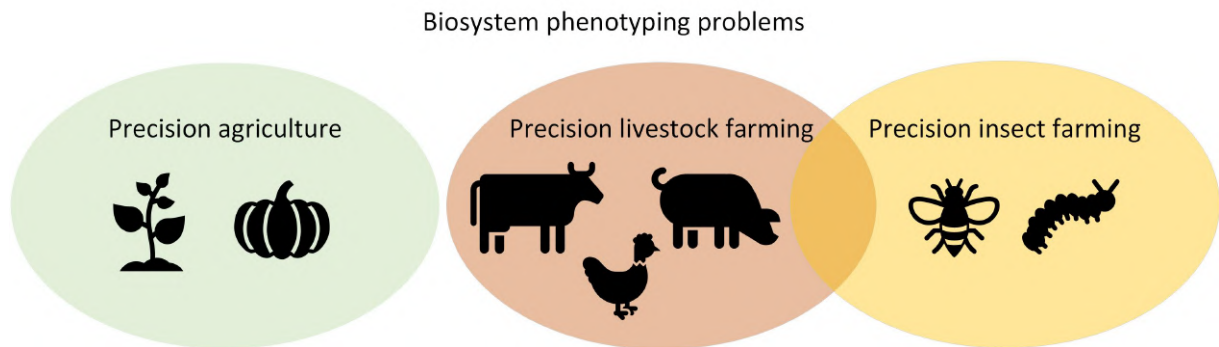


Figure 1.1: Application areas for biosystem phenotyping problems.

The potential to increase food production is also seen in new food sources. Novel food includes edible insects that can be used as food or for high-protein feed. Among the most important insect species farmed for food and feed are *Tenebrio molitor* (also known as mealworm) and *Hermetia illucens*.

Phenotyping of biosystems requires appropriate sensors (including cameras) to record the important signals. RGB imaging and acoustic signal[19, 18] registration are most widely used for phenotyping biosystems due to the relatively low cost of these solutions. Due to the need to process data in near-real time and share the results with farmers, solutions are often based on IoT systems[36, 87]. Researchers also used wearable sensors[56, 95], depth images (RGB + depth dimension)[113], IR/NIR images[52, 9], and multi-/hyperspectral images[34] for biosystem phenotyping problems.

## 1.2 Related Work

Recent advances in machine learning (ML) and computer vision (CV) have resulted in frequent use of these approaches for biosystem phenotyping problems as well. We find numerous examples of ML/CV methods for PA, PLF and PIF in the literature.

In the case of PA, ML methods have been used for (a) early detection of plant diseases[99, 103] and pests detection[111] (for precision spraying), (b) detection of weeds (for precision weeding)[51], (c) estimation of chlorophyll content[34] and water content (moisture)[120] in the plant (for precision irrigation and fertilization), (d) detection, sizing and determination of fruit maturity (for automated fruit harvesting)[21], (e) plant row position estimation[115, 68] (for in-field navigation and gaps detection), (f) yield prediction and plant growth stage

determination (for logistics planning)[110, 81], and (g) reconstruction of trees with fruit[109, 59] (for optimal cutting point determination and automated fruit harvesting).

For PLF, the following application areas of ML/CV methods should be distinguished: (a) recognition of basic activities (lying, eating)[56, 101] and behavioural patterns detection (e.g., playing [46], pawing[22]), (b) disease detection (e.g. lameness[114, 63, 119]), (c) animal welfare determination [45] with stress detection (e.g., heat stress[93]), (d) estimation of size parameters and biomass[113, 15], and (e) estimation of respiration rate (for anomaly detection)[94, 102].

The dissertation addressed PIF issues in the context of two insect species: the honeybee (*Apis mellifera*) and the mealworm (*Tenebrio molitor*). For these biosystems, it was decided to present a detailed state-of-the-art with particular attention to ML/CV methods used in each article. The objects of analysis on which the dissertation focused are shown in Figure 1.2.



Figure 1.2: Insect species considered in the dissertation: (a) the honeybee (*Apis mellifera*), and (b) the mealworm (*Tenebrio molitor*).

The papers related to the use of ML/CV methods for phenotyping honeybee and mealworm biosystems are summarized in Table 1.1. It was decided to list only studies based on images as a source of information since computer vision methods were the subject of the dissertation.

Table 1.1: Articles that used ML/CV methods to phenotype honeybee (HB) and mealworm (MW) biosystems.

article	insect	task	ML/CV methods
(Stojnić et al., 2018)[85]	HB	(1) detection of honey bees bearing pollen	(1) colour space conversion, and K-Means for segmentation bee/background, (2) PCA to decorrelate RGB channels, (3) SIFT[54], VLAD[39], and SVM[23] for image (ROI) classification
(Bozek et al., 2018)[13], (Bozek et al. 2021)[14]	HB	(1) bee detection and classification, (2) brood cell detection, (3) bee tracking (research especially aimed for dense scenes)	(1) multioutput modified U-Net[80], (2) orientation represented by continuous values in ground truth for semantic segmentation

(Rodríguez et al., 2018a)[78], (Rodríguez et al., 2018)[76], (Rodríguez et al., 2018b)[79], (Rodríguez et al., 2022)[77],	HB	(1) bee body parts detection, (2) pollen loads detection, (3) pose estimation, (4) tracking, (5) detection of entrance and exit events,	(1) Part Affinity Fields[16] method with VGG-19[82] as a feature extractor for pose estimation, (2) shallow CNN for pollen loads detection, and (3) Hungarian algorithm[44] for tracking
(Marstaller et al., 2019)[57]	HB	(1) insect species classification (bee, wasp, bumblebee, hornet), (2) bee classification (normal, with pollen, drone, dead), (3) pose estimation, (4) pollen localization	(1) multi-task architecture with shared encoder, (2) additional decoder for semi-supervised learning (train on unlabeled images)
(Ngo et al., 2019)[67]	HB	(1) counting bees entering and leaving the hive, (2) tracking bees	(1) Hungarian algorithm[44] and Kalman filter for tracking[84]
(Bjerge et al., 2019)[9]	HB	(1) <i>Varroa destructor</i> mite detection	(1) customized CNN model with inference on R-NIR-B images
(Westwańska and Respondek, 2019)[106]	HB	(1) bee counting	(1) U-Net, (2) ground truth generated based on point annotations (as circles)
(Alves et al., 2020)[3]	HB	(1) detection and classification of cells in the bee frame	(1) Circle Hough Transform[27] for cells detection, (2) MobileNet[37] for cells classification
(Dembski and Szymański, 2020)[25]	HB	(1) bee detection	(1) weighted clustering of bounding box proposals
(Tausch et al., 2020)[88], (Borlinghaus et al. 2023)[11]	HB	(1) re-identification	(1) triplet loss for training re-ID model based on embeddings extracted from ResNet-18[35] model
(Ngo et al., 2021a)[65]	HB	(1) bee detection and classification (into pollen and non-pollen)	(1) YOLOv3 model[75]
(Ngo et al., 2021b)[66]	HB	(1) bee colony daily loss rate forecasting, (2) anomaly detection for early warning	(1) temporal CNN[47] for forecasting, (2) Isolation Forest[53] algorithm for anomaly detection
(Ratnayake et al., 2021a)[73], (Ratnayake et al., 2021b)[74], (Ratnayake et al., 2023)[72]	HB	(1) bee detection and tracking for pollination assessment, (2) bee-flower interaction detection	(1) hybrid approach with background subtraction (foreground/background segmentation) and YOLOv4[10] for bee detection, (2) proposed Polytrack algorithm based on multistage processing



---

(Chan et al., 2022)[20]	HB	(1) re-identification	(1) triplet loss for training re-ID model based on embeddings extracted from a customized model with ResNet units, (2) self-supervised training of re-ID model with automatically annotated images using tracking
(Tausch et al., 2023)[89]	HB	(1) bee detection, (2) pollen loads detection, (3) pollen colour clustering	(1) U-Net for bee detection and pollen loads detection, (2) HDBSCAN[58] algorithm for pollen colour clustering based on images converted to LAB colour space
(Sledević and Plonis, 2023)[83]	HB	(1) bee detection, (2) bee tracking, (3) occurrence density maps estimation	(1) YOLOv8[40] for bee detection, (2) ByteTrack[116] for tracking, (3) maps calculated based on bee tracks
(Baur et al., 2022)[5]	MW	(1) segmentation, classification and sizing of larvae segments	(1) watershed algorithm for larvae segments segmentation, (2) MLP for segments classification
(Papadopoulos et al., 2024)[69]	MW	(1) larvae detection	(1) YOLO-NAS[1] and YOLOv8

---

The articles collected in Table 1.1 for PIF and listed in the previous paragraph (for PA and PLF) confirm that researchers have often applied ML/CV methods in the area of PIF, PA, and PLF. Selected phenotyping problems for insect biosystems are shown in Figure 1.3.

Concentrating on the methodological part and the efficiency of the whole pipeline of developing ML/CV methods for biosystem phenotyping (from developing the dataset to maintaining the model during production), we can find research gaps. First of all, it should be emphasised that the works listed involve only a selected part of the ML/CV method development pipeline, focusing on proposing a suitable method and its evaluation under the assumption of having a representative dataset. Researchers have omitted the problems of efficient acquisition strategy, sample selection, labelling and maintaining high model accuracy during production. Especially in the case of insect biosystems, these pipeline elements are particularly important. The images acquired represent dense scenes, which results in significant sample labelling time and longer image processing times during inference. Insect biosystems are also time-varying, which favours a reduction in model accuracy during production caused by domain shift and concept drift effects. In the case of the honeybee, researchers have already developed dedicated analysis methods for selected problems, but this can not be said for the mealworm, which is a challenging object of analysis and requires different approaches due to dense scenes. The referenced works mainly involved phenotyping at the population level, i.e. without focusing on the characteristics of individual specimens. The communicated research gaps are highlighted and described in more detail in the next chapter.

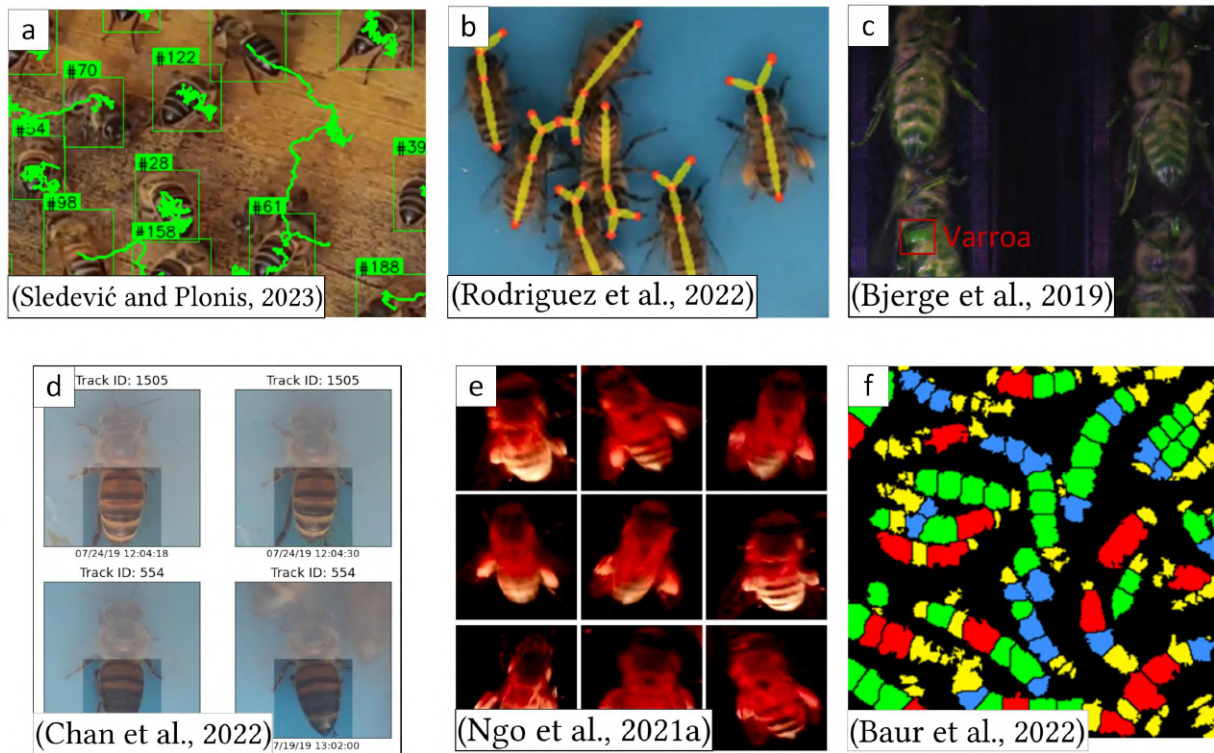


Figure 1.3: Selected phenotyping problems of insect biosystems from the literature for the honey bee: (a) bee detection and tracking, (b) pose estimation, (c) detection of the Varroa mite, (d) re-identification, (e) pollen loads detection and the mealworm: (f) larval segment separation and classification.

### 1.3 Research Gaps

In this chapter, research gaps are pointed out and discussed.

#### Research Gaps:

**[RG1]** Methods for developing representative datasets for problems characterized by the difficulty of obtaining valuable samples or the long time required to annotate samples

**[RG2]** Methods dedicated to processing images representing dense scenes

**[RG3]** Methods for supervision and adaptation of machine learning models during production

[RG4] Methods specifically dedicated to biosystem phenotyping problems, also taking into account domain knowledge (as opposed to off-the-shelf solutions)

[RG5] Methods to reduce solution complexity and inference time

[RG6] Phenotyping methods at the individual level

In the following paragraphs, further observed research gaps are discussed along with related works.

[RG1] The problem of weakly represented datasets is referred to when the available samples are not sufficient to satisfactorily reflect the variance of all potential samples for the problem posed and thus training accurate and robust machine learning model on such a dataset is difficult. The opposite of a weakly represented dataset is a representative dataset, which contains a sufficient number of samples with adequate variation. In the case of phenotyping of biosystems the problem of weakly represented datasets can be well illustrated by the example of problems of anomaly detection or pest detection. Obtaining positive samples (samples with an anomaly or images with a pest) for such problems is difficult. For some problems, the frequency of occurrence of positive samples can be less than one percent. Under such conditions, manual inspection of all samples obtained is not possible. Researchers often omit the description of the method of developing representative datasets in articles, although this is a crucial element for the success of the entire study. Among the partially satisfactory solutions, one can mention the special preparation of samples with anomalies that form the basis for the dataset. This approach was used in [9], where bee colonies with increased parasite infestation rates were prepared. With the described approach, the researchers make the risky assumption that a representative dataset can be obtained using only samples obtained in the absence of parasites and with high parasite infestation. However, it seems reasonable to assume that the nature of the data acquired under conditions between these extremes (low and medium infestation) may differ. It should be considered that with the described approach, the representativeness of the final dataset is at least questionable.

An alternative approach is to rely on a small set of labelled samples and further expand this dataset using augmentation techniques and semi-supervised methods. Augmentation involves generating new samples based on available real samples. Semi-supervised learning relies on the inclusion of unlabelled samples in subsequent stages of machine learning model development, including training and maintenance using adaptation.

In the literature, in addition to standard augmentation methods related to simple image transformations, i.e. geometric transformations (translations, rotations), change of brightness, contrast and colour, we can also find more advanced methods, namely style transfer[70], image generation using GAN[55] and patch-based augmentation (mosaic augmentation)[24]. The rationale for using style transfer depends on the problem. In the case of GAN, the significant computational cost coupled with the lack of certainty of significant performance improvement (compared to standard augmentation approaches) limits the rationality of using

this approach for augmentation[8]. Patch-based augmentation is a valuable technique, readily used in publicly available Python libraries such as YOLOv8[40]. In its original form, it involves creating an orthomosaic composed of rectangular slices from the original image. In the literature, we also find a further development of the patch-based augmentation method in the form of generating new images based on the objects extracted from the images for instance segmentation problems[92]. Another approach to augmentation is to simulate real scenes and generate synthetic images based on them [4]. Based on domain knowledge, researchers attempt to reproduce real objects, and the quality of this reproduction determines the model's effectiveness in real-world conditions. A limitation of simulation-based methods should be given the high time consumption in the preparation of simulations and the problem of domain shift between images derived from simulations and real images[60]. The augmentation methods described are shown in Figure 1.4

Improvements in model performance using augmentation techniques are limited because they are based on an initial set of samples and may not be sufficient in the case of a weakly represented initial dataset. In that case, further performance improvement should be sought in the inclusion of unlabeled image pools and semi-supervised methods in training.

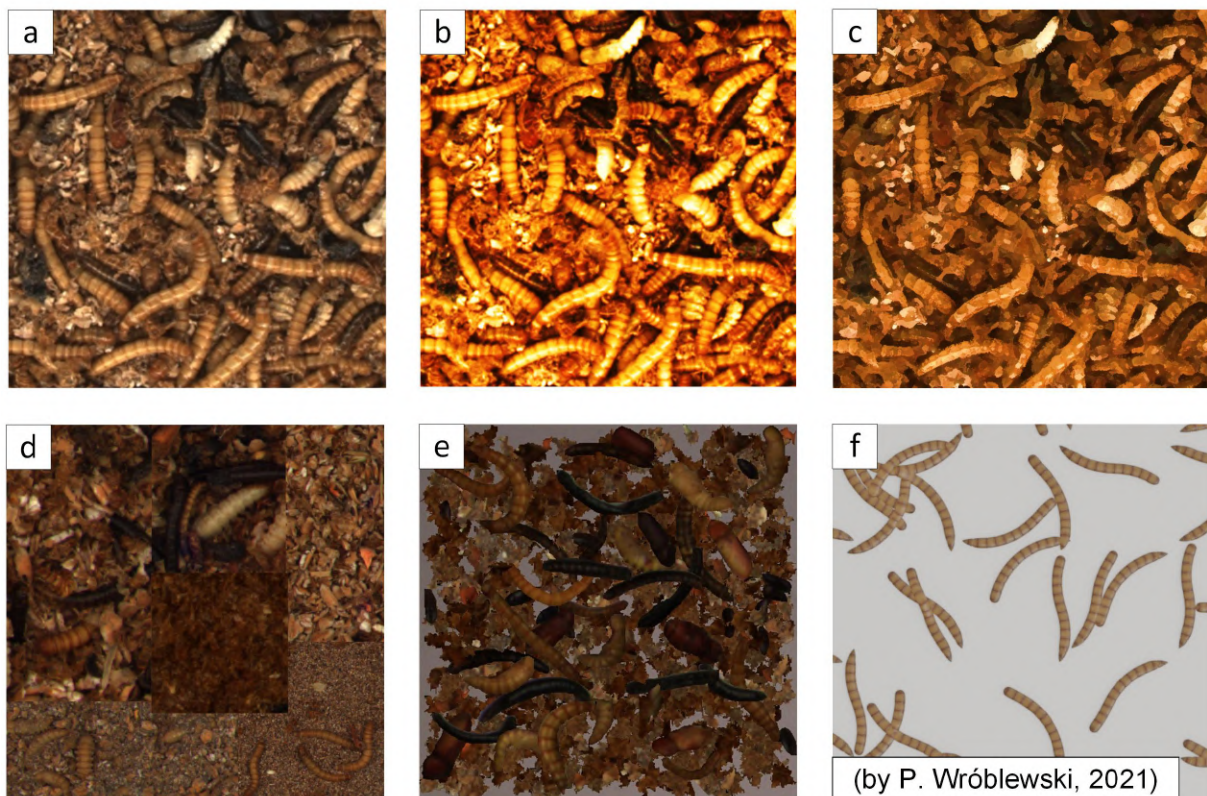


Figure 1.4: Selected augmentation methods: (a) original RGB image, (b) contrast and brightness modification, (c) style transfer, (d) patch-based augmentation (mosaic), (e) generated synthetic images based on extracted objects, and (f) simulation-based.

The pseudo-label-based self-training approach[38] is one example of a semi-supervised



approach that can be useful in the development stage of machine learning models. The method involves determining a pseudo-label (also referred to in the literature as a noisy label or pseudo-ground truth) as a prediction of a previously trained (so-called weak model) on a labelled set of samples. An appropriate confidence score threshold is selected to obtain the labels (ground truths) from the predictions associated with the confidence score. In the literature, we also find expansions of the pseudo-labelling method mainly based on additional methods for refinement of pseudo-labels[42]. Pseudo-labels can also be corrected using domain knowledge. Figure 1.5 shows an idea scheme for pseudo-label-based self-training with incorporating domain knowledge. In the context of including unlabeled samples for model fine-tuning, the approach presented in the article [57] is also worth noting, where an architecture with a shared backbone as an encoder and a separate branch for handling unlabeled data was used. An additional branch was the decoder, and the multi-scale structural similarity (MS-SSIM) proposed in [105] was used to train it.

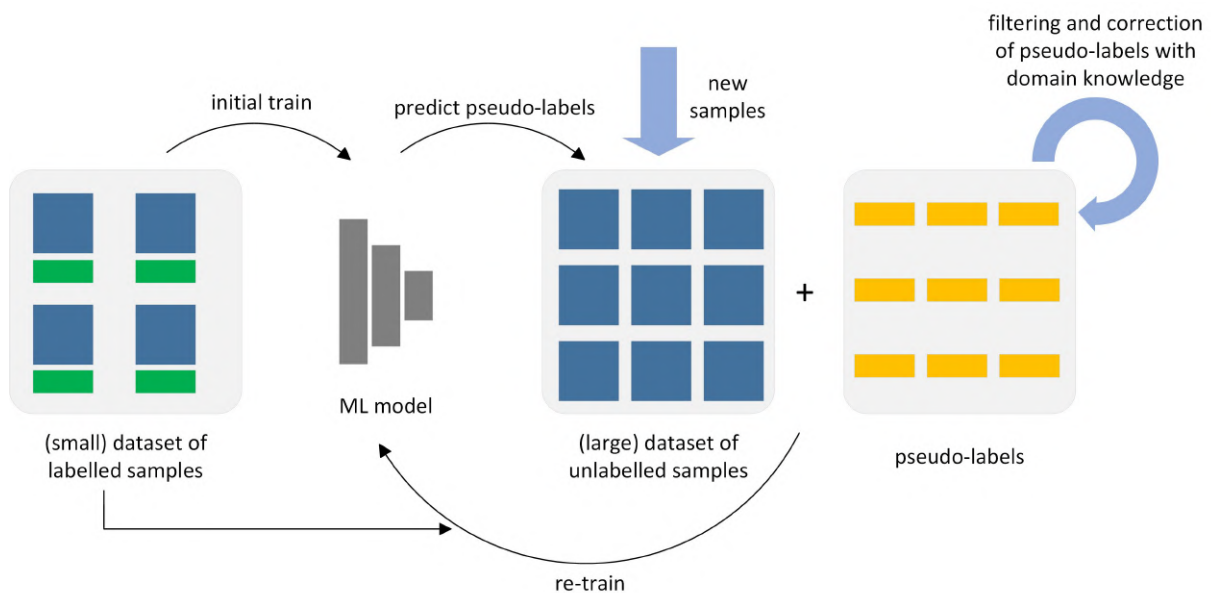


Figure 1.5: Idea of pseudo-label-based self-training with incorporating domain knowledge.

**[RG2]** The problem of dense scenes occurs when there are many objects in the image, causing significant overlapping of objects and difficulty in counting them. Dense scenes are another problem in developing representative datasets, mainly due to the long time spent on labelling. The pseudo-labelling method, described in the previous paragraph, can speed up labelling. It can also be supported by manual inspection to avoid including incorrectly determined pseudo-labels in training and further propagation of the error. In the context of dense scenes, it is worth noting that approaches based on simplified labels, such as replacing polygons or bounding boxes with point annotation or skeleton, significantly reduce the annotation time investment. In [106], point annotation was used to train a U-Net model for bee detection. We can also see a similar approach in other application areas [96, 33]. However, it should be noted here that point annotation also has limitations. This approach is not optimal

when it is impossible to propose a central point for an object (elongated objects such as larvae) and when objects are frequently overlapping, as the central point may be covered. Another issue is the problem of the adopted labelling strategy in dense scenes when there is a problem of overlapping. The standard approach is based on labelling only the visible parts of objects. However, for some problems, it may be required to reconstruct the area of invisible parts of the object, which is handled by amodal segmentation[49].

**[RG3]** Insect biosystems are dynamic and changeable, so it must be assumed that domain shift and concept drift effects occur during production time for the developed machine learning models. The domain shift effect refers to the difference in sample characteristics between the source domain on which the model was trained and the target domain in which the model is to operate and is being evaluated. In the case of concept drift, we refer to a continuous change in the distribution of samples over time. The domain shift effect for the mealworm biosystem phenotyping problem is presented in Figure 1.6. In Figure 1.6 representing the TSNE results on the extracted deep features, we can see the separation of the grouped samples representing different domains. The majority of work for this problem in literature is related to domain adaptation methods (e.g. self-supervision-based[117], adversarial-based[118]). A typical scenario is to try to adapt the model to a new dataset whose samples come from an acquisition using a different vision system. Of course, we can also find such problems for PIF when the model needs to be implemented in new farming conditions. This should be treated as a one-time activity. More important from an application point of view is the maintenance of the model during production with the detection of changes in the nature of the data and the estimation of prediction uncertainty. Semi-supervised approaches, especially self-training[42], are worth considering in the context of maintaining machine learning models for monitoring insect biosystems. The advantage of this approach is that it is relatively easy to incorporate domain knowledge into the adaptation procedure by proposing a suitable pseudo-label correction method. The problem of estimating prediction uncertainty can be solved by using model ensemble and bootstrapping, which is possible for models with relatively low inference time. The problem of reducing model complexity was addressed in [RG5].

**[RG4]** Before the widespread use of CNNs (representing automatic feature extractors)[43] approaches based on feature engineering and classical machine learning models, i.e. SVM[23], random forest, played a significant role in computer vision. The undoubted advantage of such approaches was the relatively easy explainability of inference since the proposed features were based on prior knowledge. Among the classical approaches, it is worth noting the methods based on ontology[86]. Of course, CNNs tend to perform better than classical approaches, assuming the development of a representative (with a sufficiently large number of samples) dataset for model training, but automatic extractors should not be considered capable of learning all problem-relevant patterns. Hybrid approaches, combining state-of-the-art methods from the ML area with prior knowledge, make it possible to increase the accuracy of the solutions being developed while increasing the explainability. The described approaches to computer vision problems are summarised in Figure 1.7. In the literature, we find a couple of solutions based on hybrid approaches[98, 108], but this is a much smaller number of articles than the number of papers using off-the-shelf models in either standard form or improved[50,

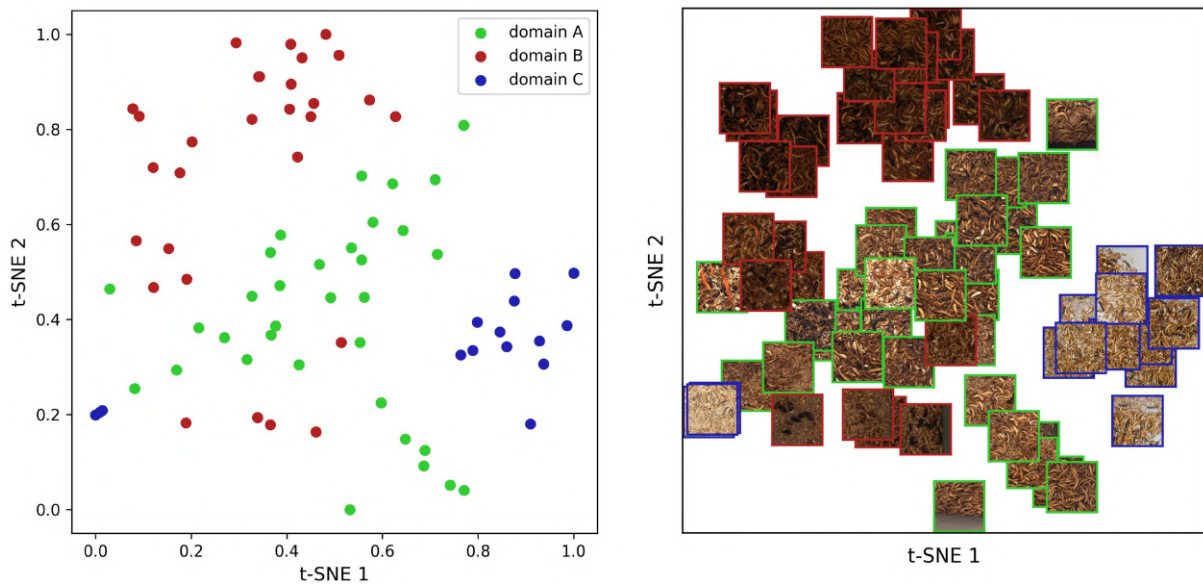


Figure 1.6: Visualization of the domain shift effect with TSNE analysis using features extracted from the ResNet model.

91]. An important direction is problem-oriented architectures, often multi-purpose in the form of end-to-end solutions[57, 77]. They tend to outperform benchmark approaches (e.g., vanilla YOLO or Mask R-CNN) with reduced inference time due to not having to apply complex architectures to specific problems.

**[RG5]** The preference for low-complexity solutions is related to the need to provide near real-time inference. Recently, one can also see an increase in researchers' interest in the energy consumption of solutions and environmental costs[41]. The first step to reduce complexity is to choose model architectures with as low complexity as possible, providing acceptable performance. Popular Python libraries, i.e. YOLOv8[40] give users a choice of architectures of varying complexity. Another approach is the knowledge transfer between a more complex solution and a simplified one. The knowledge transfer method can take many forms. A frequently used approach is knowledge distillation[100] e.g. teacher-student learning[7] based on training a smaller architecture (named as a student) on the outputs of a larger architecture (named as a teacher). We also find approaches where training of the target architecture is performed on samples whose labels were obtained using multi-stage processing using classical computer vision methods[6] or automatic segmentation methods[17] or clustering[26]. Another group of approaches are end-to-end architectures that provide a multi-task solution within a multi-output architecture. In the context of end-to-end architectures, three issues are worth noting, namely (a) multi-output regression models[113], (b) extended base architectures using additional heads (branches)[15, 107], and (c) multi-task learning[104]. For some problems, the output is a certain number of numerical indicators. Due to the homogeneity of the outputs for these types of problems, multi-output regression models trained using standard MSE metrics can be used. The minimized training loss then takes into account the subsequent MSE metrics associated with the subsequent problems taken. In the

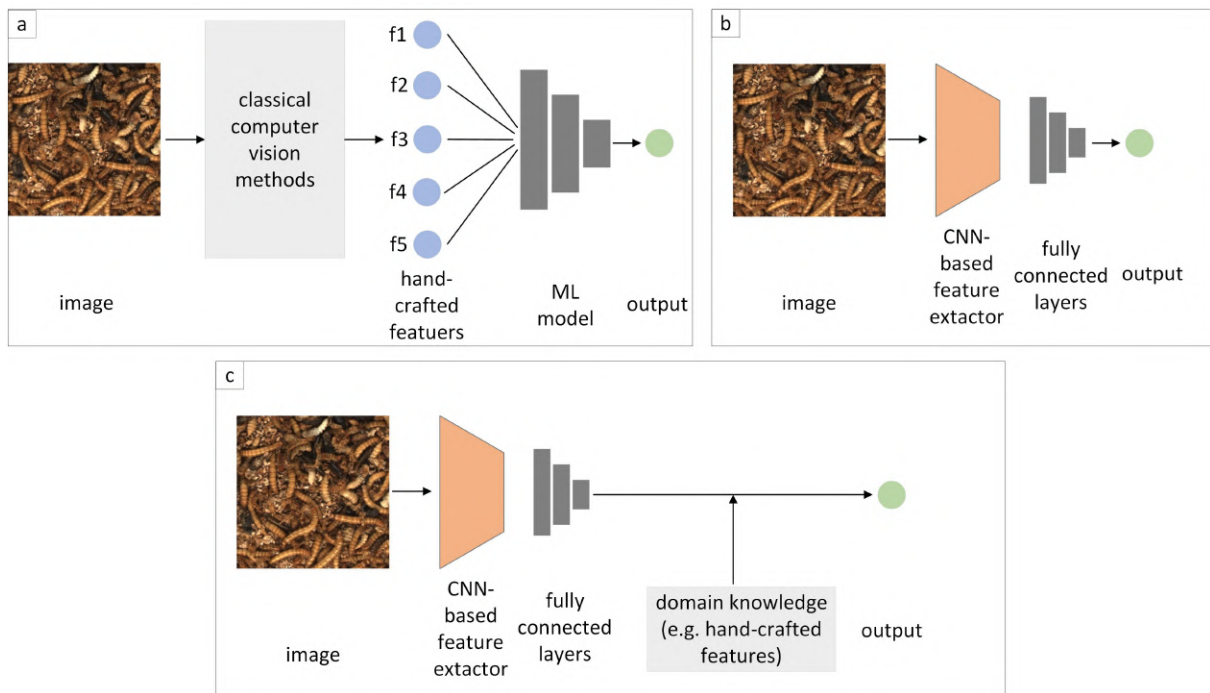


Figure 1.7: Selected approaches to computer vision problems: (1) feature engineering and classical machine learning models, (2) feature extractors (e.g., CNN-based), and (3) hybrid approaches (feature extractors and incorporation of domain knowledge).

case of multi-task learning, the output is heterogeneous, so for example, we need to determine a numerical indicator (regression task), bounding boxes (object detection task) and a mask (semantic segmentation task). The challenge with multi-task learning is to propose a suitable loss for training combining the defined tasks. A very important and useful approach is to extend standard architectures, i.e. Faster R-CNN[15] or YOLO[107] with new heads (branches) for new tasks. The new heads can then be trained separately or in multi-task learning mode. The types of end-to-end architectures discussed are presented in Figure 1.8

**[RG6]** For insect biosystem phenotyping problems, the dominant approach is to calculate indices that characterize the entire population, which in most cases is a reasonable approach, considering dense scenes. An example of population-level phenotyping is counting all bees entering and exiting the hive at the entrance to the hive[65]. In the case of individualized phenotyping, we would analyze the entrances and exits of specific bees with IDs from the hive. In the case described, individualized phenotyping based on images is very difficult to implement due to the high similarity between individuals and the tens of thousands of individuals in the colony. Similar studies are carried out using RFID tags[31]. Another example with a comparison of phenotyping at the individual and population level on the example of the mealworm is shown in Figure 1.9.

We can also find problems for which phenotyping at the level of individuals is crucial, e.g. selective breeding studies[61], where the usefulness of individuals for further reproduction is assessed. For long-term phenotyping, re-identification of individuals is needed, which



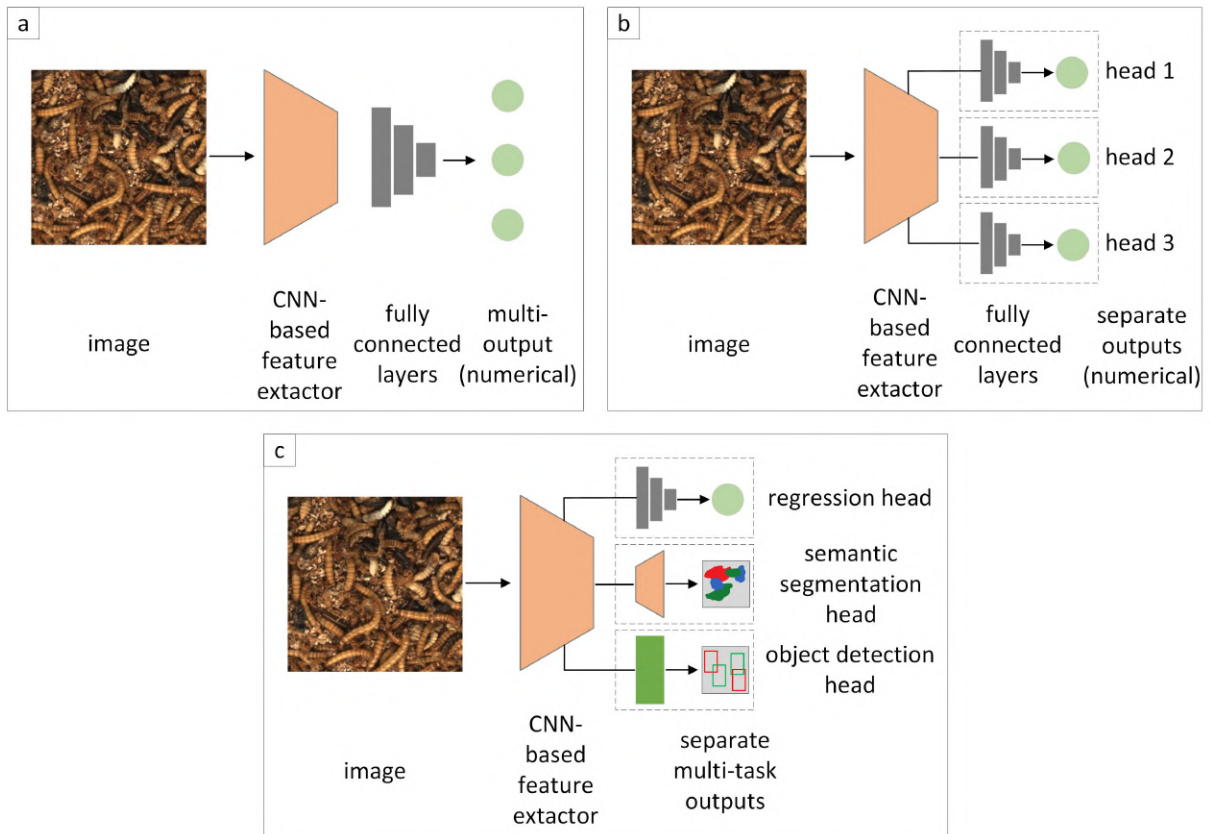


Figure 1.8: Types of end-to-end architectures: (a) multi-output regression models, (b) extended base architectures using additional heads, and (c) multi-task architecture.

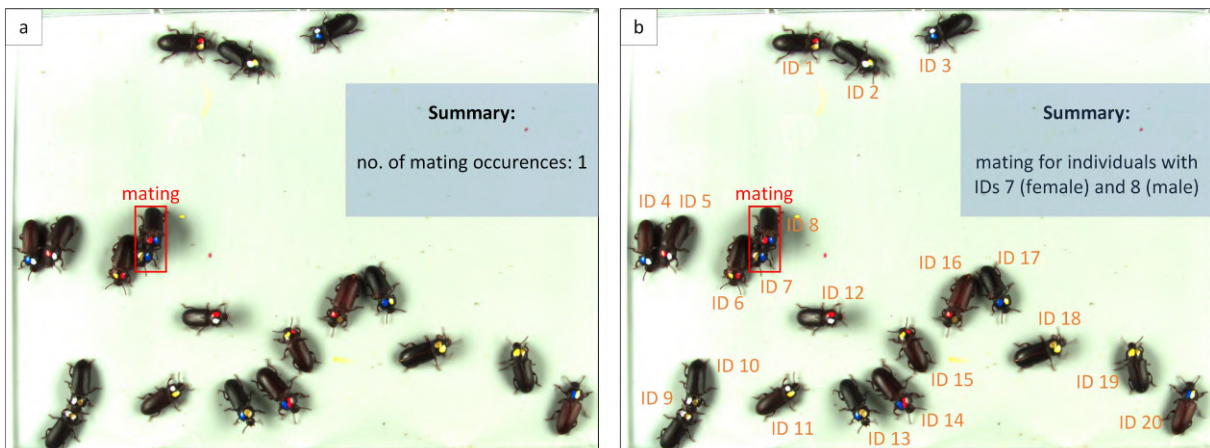


Figure 1.9: Comparison of phenotyping at the level of: (1) population, and (2) individuals for the mealworm beetles.

makes it possible to identify the same individuals on different frames and link relevant dynamic behavioural patterns to them. In the literature, we find the first works related to

the re-identification of insects[62, 11, 20]. However, they should be considered as preliminary studies since (1) they are not set in the context of specific application problems and (2) for the selected works[62], the chosen methods of validation under partially laboratory conditions are not sufficient to conclude that re-identification under real conditions will be effective.

## 1.4 Research Hypothesis and Objectives

Considering the state of current knowledge along with the identified research gaps, we defined the following research hypothesis and objectives.

### Research Hypothesis:

Machine learning methods using synthetic images, semi-supervised learning, knowledge transfer and end-to-end architectures enable the development of dedicated models for phenotyping insect biosystems that are more efficient, easier to develop and maintain and characterized by shorter inference times than currently used machine learning methods.

### Research Objectives:

[O1] Development of method enabling faster development of machine learning methods for phenotyping insect biosystems, involving synthetic image generation and semi-supervised learning (pseudo-labeling)

[O2] Development of method enabling more efficient maintenance during the production of machine learning methods for phenotyping insect biosystems, involving detecting domain shift (or concept drift) effect and adaptation technique

[O3] Development of method enabling reduction of complexity (inference time) of machine learning methods for phenotyping insect biosystems, involving knowledge transfer and end-to-end model

[O4] Development of method enabling the incorporation of domain knowledge (a priori) in the development, maintenance, and inference of machine learning methods for phenotyping insect biosystems

[O5] Development of method enabling phenotyping insect biosystems at the level of individuals (rather than population), involving re-identification and detection of behavioural patterns



# CHAPTER 2

## Results Summary

---

This chapter presents a summary of the most important achievements for the dissertation, i.e. a list of publications, listed other important achievements, and providing justifications for achieving the set research objectives.

### 2.1 Publications

The doctoral dissertation consists of the following seven articles (six published or accepted for publication and one under review):

[A1] Majewski, P., Zapotoczny, P., Lampa, P., Burduk, R., & Reiner, J. (2022). Multipurpose monitoring system for edible insect breeding based on machine learning. *Scientific Reports*, 12(1), 7892.

**DOI:** 10.1038/s41598-022-11794-5

**Publication status:** published

[A2] Majewski, P., Lampa, P., Burduk, R., & Reiner, J. (2023). Prediction of the remaining time of the foraging activity of honey bees using spatio-temporal correction and periodic model re-fitting. *Computers and Electronics in Agriculture*, 205, 107596.

**DOI:** 10.1016/j.compag.2022.107596

**Publication status:** published

[A3] Majewski, P., Mrzygłód, M., Lampa, P., Burduk, R., & Reiner, J. (2024). Monitoring the growth of insect larvae using a regression convolutional neural network and knowledge transfer. *Engineering Applications of Artificial Intelligence*, 127, 107358.

**DOI:** 10.1016/j.engappai.2023.107358

**Publication status:** published

[A4] Majewski, P., Lampa, P., Burduk, R., & Reiner, J. (2023). Mixing Augmentation and Knowledge-Based Techniques in Unsupervised Domain Adaptation for Segmentation of Edible Insect States. *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, 380–387.

**DOI:** 10.5220/0011603500003417

**Publication status:** published

[A5] Majewski, P., Lampa, P., Burduk, R., & Reiner, J. (2024). Improved Pest Detection in Insect Larvae Rearing with Pseudo-Labeling and Spatio-Temporal Masking. *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 2: VISAPP*, 349–356.

**DOI:** 10.5220/0012311300003660

**Publication status:** published

[A6] Majewski, P., Lampa, P., Burduk, R., & Reiner, J. (2024). End-to-end Solution for *Tenebrio Molitor* Rearing Monitoring with Uncertainty Estimation and Domain Shift Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5498-5507)*.

**Publication status:** accepted for publication as CVPR (Conference on Computer Vision and Pattern Recognition) 2024 workshop paper

[A7] Majewski, P., Lampa, P., Burduk, R., Reiner, J., & Lin, T.T. (2024). Phenotyping with dynamic characteristics determination for *Tenebrio Molitor* beetles in selective breeding using re-identification.

**Publication status:** submitted to *Engineering Applications of Artificial Intelligence*

Other details regarding the listed publications were presented in Chapter 4.

## 2.2 Other Important Achievements

In this chapter, I also wanted to list other articles that I did not include directly in the dissertation:

**[A8.supp]** Majewski, P., & Reiner, J. (2022). Hybrid Method for Rapid Development of Efficient and Robust Models for In-row Crop Segmentation. Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 4: VISAPP, 274–281.

**DOI:** 10.5220/0010775400003124

**Publication status:** published

**[A9.supp]** Marciniak, K., Majewski, P., & Reiner, J. (2024). Estimation of the Inference Quality of Machine Learning Models for Cutting Tools Inspection. Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: VISAPP, 359–366.

**DOI:** 10.5220/0012321900003660

**Publication status:** published

During the realization of my PhD, I also had other achievements that I would like to mention to characterize my multifaceted scientific activity:

- (1) research internship at the Biophotonics and Bioimaging Laboratory at the National Taiwan University under the supervision of Professor Ta-Te Lin (four months, 08.2023 - 11.2023),
- (2) research leader in the project 'Automatic mealworm breeding system with the development of feeding technology' (NCBiR, grant POIR.01.01.01-00-0903/20, 01.04.2021 - 31.05.2023),
- (3) cooperation with Tenebria (Lubawa, Poland) company within the project 'Automatic rearing system *Tenebrio molitor* with the development of feeding technology',
- (4) patent application 'Method of monitoring the rearing of edible insects' (20% contribution, submitted 08.2023),
- (5) machine learning and computer vision specialist in four R&D projects carried out at the Wrocław University of Science and Technology,
- (6) preparation of project proposals for national (NCN Preludium, NCBiR) and European (Horizon) contests,
- (7) one oral and four poster presentations at international conferences (CVPR, VISIGRAPP, ECPA),
- (8) five oral and one poster presentations at national conferences,

- (9) scholarship of the president of Wrocław for multidisciplinary research (12.2023),
- (10) reviewer for the journals: *Insects* (ISSN 2075-4450), *Remote Sensing* (ISSN 2072-4292).



## 2.3 Achieving Research Objectives

The purpose of this chapter is to confirm the achievement of the research objectives in the context of the publications on which the dissertation is based. For each research objective, relevant publications are identified and the contribution of the publications to the research objective is described.

**[O1]** Development of method enabling faster development of machine learning methods for phenotyping insect biosystems, involving synthetic image generation and semi-supervised learning (pseudo-labeling)

**Relevant publications:** A1, A3, A4, A5, A6, A7

**Description:**

In article [A1] I proposed a method for generating synthetic images based on pools of objects from different classes. The method developed consisted of placing successive objects on the background image and determining the ground truth at the end of the image generation process for instance segmentation (Mask R-CNN) and semantic segmentation (U-Net) tasks. Object pools could be completed by extracting objects from images based on manual annotations (this approach was used in [A1] and [A3]) or automatically (specially prepared samples with separated individuals and classical image processing methods for background removal, and filtering). It was confirmed that using only synthetic images during training of instance segmentation and semantic segmentation models makes it possible to obtain models with satisfactory accuracy, meeting the requirements of the monitoring system of mealworm rearing. Using synthetic images enabled significantly reduce model development time by avoiding the need for full labelling of images representing dense scenes. The annotator could select only the most important objects to annotate in the image. The proposed method of generating synthetic images allows to control of the density of objects in the image and their degree of coverage, resulting in the ability to simulate real dense scenes and reduce the problem of imbalance in the number of objects from certain classes. The described pool-based 2D synthetic image generation method is presented in Figure 2.1. As a distinctive characteristic (novelty) of our method of generating synthetic images from other works[92, 30], the assumption of multi-class (eight classes in the version shown in Figure 2.1) and multi-task (instance segmentation and semantic segmentation) should be emphasized.

The article [A4], in an additional experiment, checked the loss of accuracy when training an instance segmentation model (classes live larva, dead larva, pupa) only on synthetic images in comparison to training the model only on real images and on a mix of synthetic and real images. The analysis was performed for in-domain and out-domain inference (after applying the proposed domain adaptation method). A significant difference was observed in the accuracy of the models, especially for in-domain inference, comparing different strategies for developing the training set. The main conclusion of this experiment was the need to consider using a model training strategy on a mix of synthetic and real images when time is available for labelling real images representing dense scenes. The strategy of training the

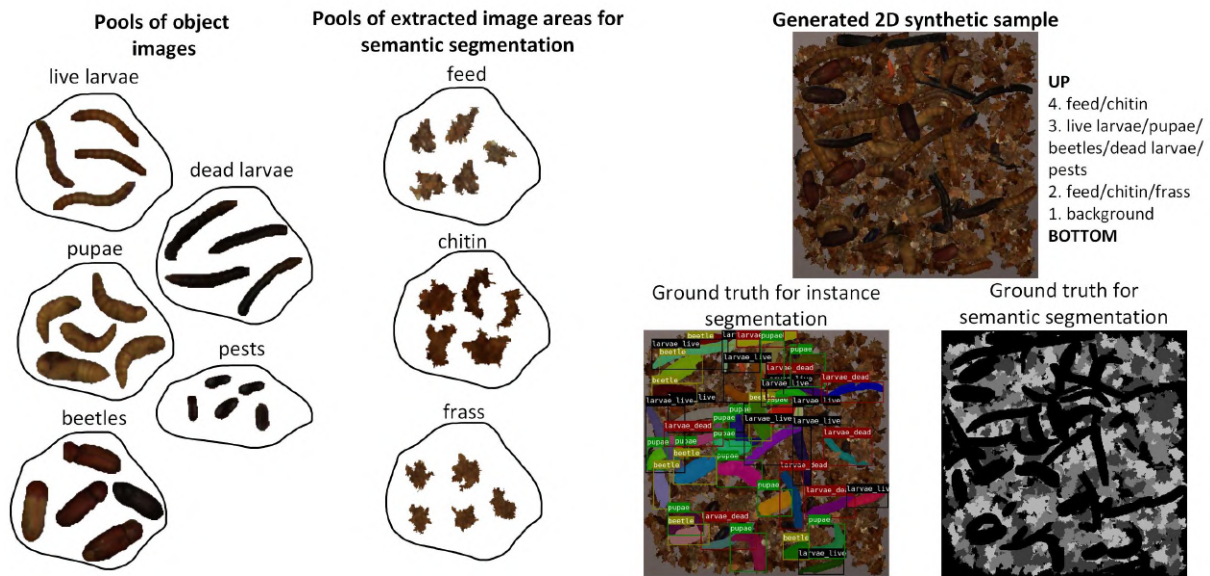


Figure 2.1: The pool-based 2D synthetic image generation method [A1].

model on a mix of synthetic and real images made it possible to increase the AP50 for object detection from 73.9 to 85.2 (for in-domain inference) and from 67.4 to 71.8 (for out-domain inference), compared with the results for the strategy of training the model on synthetic images only.

In article [A3] I proposed a 3-stage approach for developing an instance segmentation model for larvae. The developed method used the generated synthetic images to train an instance segmentation model for mealworm larvae in the first and second stages of the proposed method for developing an instance segmentation model. In the third stage, it was proposed to train the model on a set of samples consisting of synthetic images (with automatically generated ground truth) and real images (ground truth were pseudo-labels, i.e. inference results of the previously trained instance segmentation model at a certain confidence score threshold). The proposed three-stage method of developing the instance segmentation model made it possible to increase the AP50 from 75.0 (after stage one, training on synthetic images only) to 79.2 (after stage three, training on a mix of synthetic and real images). The developed method addressed the problem in the article [A4], where the problem of noticeably lower segmentation/detection performance was observed when labelled real images were not included in the training set. With the proposed method, a compromise was reached - real samples are included in the training set but labelled automatically based on the prediction of the model trained on the set of synthetic images. The described approach to developing instance segmentation models is illustrated in Figure 2.2. The originality of the proposed approach lies in combining the concept of generating synthetic images with pseudo-labelling in the successive stages of developing the instance segmentation model. The main advantage of this approach is the possibility of avoiding manual labelling entirely when the prepared samples (containing separated objects) allow the initial pool of objects to be automatically filled in.

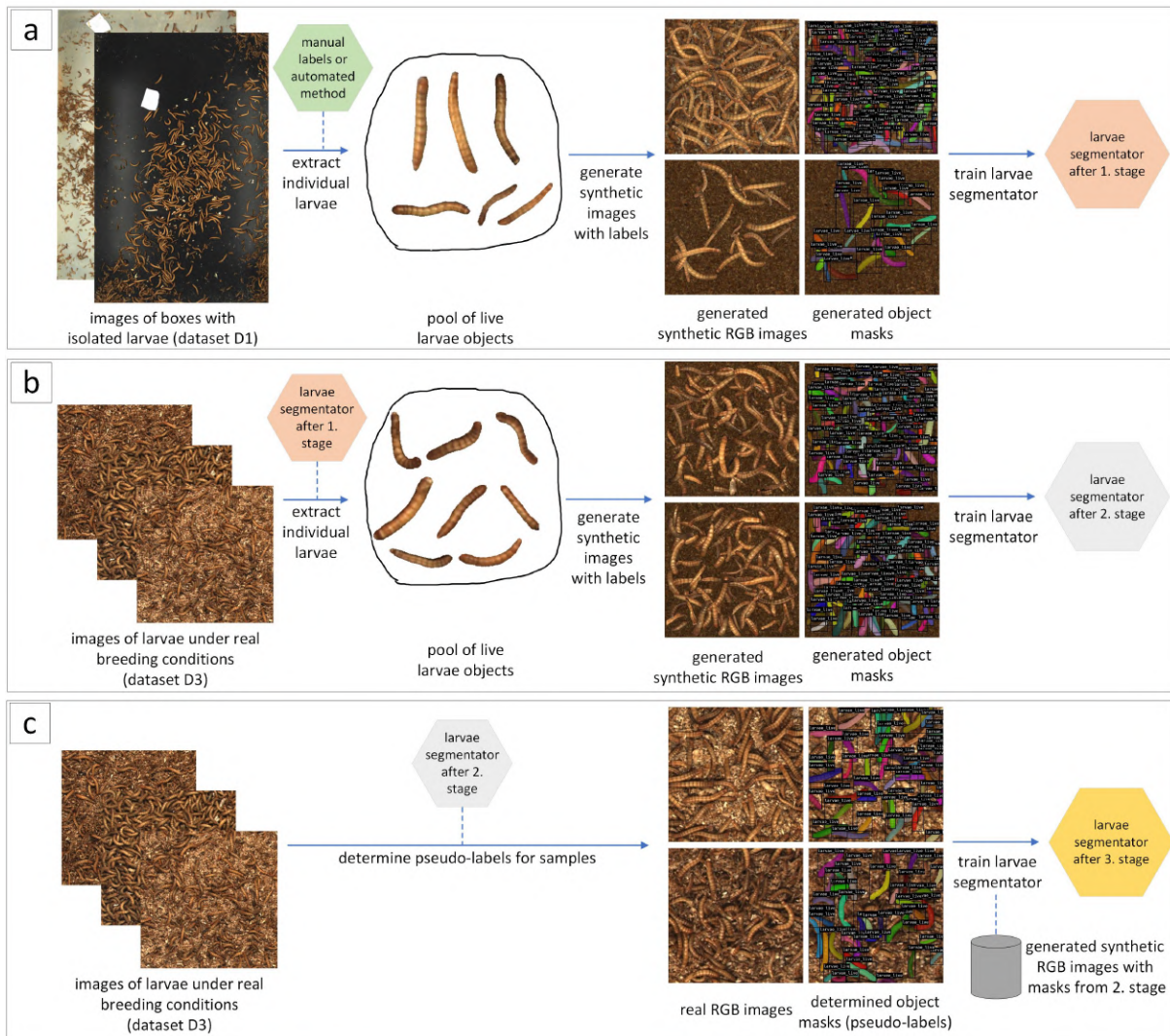


Figure 2.2: The 3-stage approach for developing an instance segmentation model for larvae [A3].

In article [A5] I proposed an improved method for developing a model for object detection (specifically for pest detection) with the problem of weakly represented datasets. The problem assumed having a small set of labelled images at the beginning (less than 100 labelled pests) and a much larger set of unlabeled images. Due to the low probability of pests in an image, even typing images for manual labelling was problematic. The proposed method first developed a small set of labelled samples using images of specially prepared rearing boxes (boxes with the pest without a pest control strategy). An initial pest detection model was then trained on the small set of labelled samples. The training was then repeated on a set consisting of images with true labels (the initial small set) and with pseudo-labels (the inference results of the initial detection model at a specified confidence score threshold). Pseudo-labels were determined for images from a set of unlabeled samples. The model obtained in this way was

used to type infected rearing boxes for further labelling and expand the set of labelled samples. It was confirmed that pseudo-labelling positively affected detection performance at different sizes of the initial set of labelled samples (more significant effect at smaller sizes). In the end, a representative set of samples for pest detection was obtained, and the possibility of using the described semi-supervised techniques to significantly accelerate the development of representative training sets of samples was confirmed. The final developed models were characterized by an F1-score of 68.6 (for inference at low/moderate pest infestation) and 82.6 (for inference at high pest infestation), with the following F1-scores for the initial detection models: 45.2 (low/moderate pest infestation) and 63.3 (high pest infestation), which is a significant improvement in detection performance. The improved method for developing an object detection model with weakly represented datasets is described in Algorithm 1.

---

**ALGORITHM 1**Improved method for developing object detection model with weakly represented datasets

---

**Input:** set of labelled images  $S \in \{s_1, s_2, \dots, s_n\}$ set of unlabelled images  $U \in \{u_1, u_2, \dots, u_m\}$ confidence score threshold  $cs_{thresh}$ number of detected objects threshold  $n_{thresh}$ **Output:** object detection model trained on extended labelled dataset

- 1: **train** model on  $S$
  - 2: **predict** bounding boxes for images from  $U$
  - 3: **filter out** bounding boxes with  $cs < cs_{thresh}$
  - 4: **filter out** bounding boxes with domain knowledge-based rules (optional)
  - 5: **re-train** model on  $S$  and  $U^*$  (with pseudo-labels)
  - 6: **repeat** steps 2-4 with re-trained model
  - 7: **count** number of detected objects  $n_u$  for images from  $U$
  - 8: **select** images from  $U$  for which  $n_u > n_{thresh}$
  - 9: **label** selected images from  $U$
  - 10: **re-train** model on  $S$  and  $U^{**}$  (with true labels)
- 

The main advantage of the proposed procedure in Algorithm 1 is the possibility to obtain relevant samples from a large amount of acquired data (e.g. from data streams), assuming a relatively low probability of a positive sample in the image.

In the article [A6], using the advantages of the proposed end-to-end architecture, I proposed a labelling strategy based on small sets of labelled samples developed for each of the defined tasks separately, e.g., a separate dataset was developed for the determination of the image coverage coefficient of the chitin, where only the chitin moults were annotated. With this approach, the most valuable images from the point of view of the problem being undertaken were selected, ensuring image diversity. The proposed approach eliminated the need to perform a complete annotation of the images (taking into account all tasks), thus reducing the overall time required for labelling.

In the article [A7] I proposed methods to efficiently develop a re-identification model of



mealworm beetles for individual phenotyping. Training sets of samples for re-identification were completed automatically (without manual labelling) by proposing a training stage in which individuals were isolated in stations while allowed to move around. During the training stage, images representing different views of the same beetle were collected. The presented acquisition procedure for the re-identification model is shown in Figure 2.3. The novelty of the proposed solution lies in providing high-quality input data (varied with true labels) for training the re-identification model by proposing an optimal sample acquisition strategy. The proposed approach offers the possibility of obtaining a much more diverse training dataset than augmentation-based dataset expansion methods.

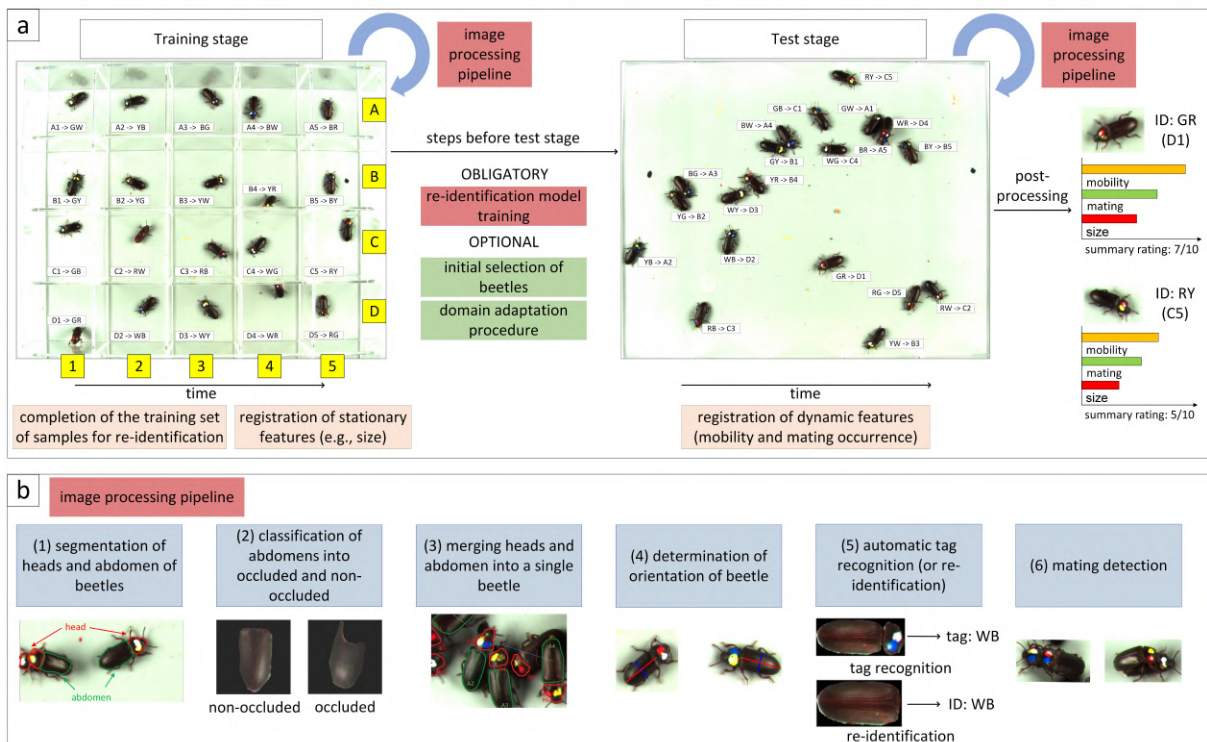


Figure 2.3: The data acquisition procedure for the re-identification model development: (a) division into training and testing stages, and (b) image processing pipeline [A7].

When individuals were highly mobile, the efficiency of the data acquisition procedure was higher. To increase the efficiency of phenotyping in the testing stage, a method for the initial selection of individuals for the testing stage was also proposed, favouring those individuals for which re-identification in the testing stage was expected to be easier. A custom metric was used to rank individuals, considering the ease of re-identification on the validation set and the variety of acquired views during the training stage. The effectiveness of the proposed method of initial selection of individuals was quantitatively confirmed. The method for calculating hybrid metrics to rank individuals for re-identification is described in Algorithm 2. The novelty in the proposed procedure in Algorithm 2 is the inclusion of sample variation in the training set in calculating the metric. To calculate the variation of samples in the training set, I proposed an original method based on SSIM coefficients.

For the mating pattern detection problem, a method of expanding the training set with additional synthetic images was also proposed, achieving an improvement in AP50 detection accuracy from 77.4 to 83.5.

---

**ALGORITHM 2**

Method for calculating hybrid metric to rank individuals for re-identification

---

**Input:** sequence of acquired images for chosen individual  $S = \{s_1, s_2, \dots, s_n\}$

$k$  - number of folds in cross-validation

$\alpha$  - coefficient determining the influence of cross-validation results on the rank score

$\beta$  - coefficient determining the influence of sample variety on the rank score

$SSIM_{thresh}$  - threshold value of SSIM to consider a change as relevant

$m_{ReID}^{thresh}$  - parameter normalising the selected metric for re-identification

$a$  - parameter normalising the number of relevant samples

**Output:** rank score for each chosen individual

```

1: divide S into  $k$  equal-sized parts                                ▷ part related to cross-validation
2: metric_values=[], j=1
3: while  $j \leq k$  do
4:   train model on parts with ids  $\neq j$ 
5:   test model on  $j$ -th part and calculate metric
6:   metric_values += metric
7:   j+=1
8: end while
9:  $coef f_1 = (\text{mean}(\text{metric\_values}) - m_{ReID}^{thresh}) / (1 - m_{ReID}^{thresh})$ 
10:
11:  $n_{relevant} = 0$                                                 ▷ part related to sample variety determination
12: i=1
13: while  $i < n$  do                                              ▷ analyse all neighbouring pairs
14:   if  $SSIM(s_i, s_{i+1}) < SSIM_{thresh}$  then
15:      $n_{relevant} += 1$ 
16:   end if
17:   i+=1
18: end while
19:  $coef f_2 = (1 - \exp(-n_{relevant}/a))$ 
20:
21: rank_score= $coef f_1^\alpha coef f_2^\beta$ 

```

---

[O2] Development of method enabling more efficient maintenance during the production of machine learning methods for phenotyping insect biosystems, involving detecting domain shift (or concept drift) effect and adaptation technique

**Relevant publications:** A2, A4, A6, A7

**Description:**

In article [A2] I addressed the problem of adapting a regression model to changes in the nature of the data during the beekeeping season. The task posed was to predict the time remaining to the end of the bees' daily foraging activity based on changes in the number of bees at the entrance to the hive and other indicators (time to sunset, environmental indicators). The approach was to periodically (once a day) re-fit the regression model using automatically pre-determined pseudo-target values. A spatio-temporal correction method based on domain knowledge (described in more detail in the description for the research objective [O5]) was used to increase the accuracy of determining pseudo-target values. The proposed periodic model re-fitting with the spatio-temporal correction method enabled a significant reduction in the RMSE prediction error compared to the reference method in the two beekeeping seasons considered (reduction from 52.5 min to 23.1 min in the 2021 season and from 71.2 min to 26.5 min in the 2022 season). The results obtained were also not significantly different from the upper baseline, that is, the results with periodic re-fitting of the model using the true target values (RMSE values of 18.5 min for the 2021 season and 27.0 min for the 2022 season). The proposed solution for predicting the time remaining to the end of the bees' daily foraging activity is shown in Figure 2.4.

In article [A4] I proposed a two-stage method for adapting an instance segmentation model (live larva, dead larva, pupa) to a new domain. The domain in the research conducted was related to a different vision system (different camera, lighting). The first stage of the proposed method was based on performing augmentation of objects in the pool, generating synthetic images and training the model on a set containing the generated synthetic images and real images from the source domain. The second stage was based on developing the object pool for the target domain and repeating the synthetic data generation and training procedure. The initial completion of the object pool for the second stage was based on the model predictions from the first stage. This was followed by knowledge-based filtering of the objects in the pool (this approach is described in more detail in the description for the research objective [O4]). Model training in stage two was carried out on a set consisting of generated synthetic images based on objects from the source domain, generated synthetic images based on objects from the target domain and real images from the source domain. The proposed two-stage domain adaptation method allowed an increase in the accuracy of the model from AP50 58.4 (without adaptation for out-domain inference) to 62.9 (after the first stage) and to 71.8 (after the second stage). The procedure for developing a two-stage domain adaptation method is additionally presented in Algorithm 3. The novelty of the proposed domain adaptation method is that it is based on the generation of synthetic images using developed pools of objects that can be easily modified using domain knowledge.

In paper [A6], with the proposed condensed end-to-end model architecture and limited inference time, I proposed to perform inference in model ensemble mode. Bootstrapping

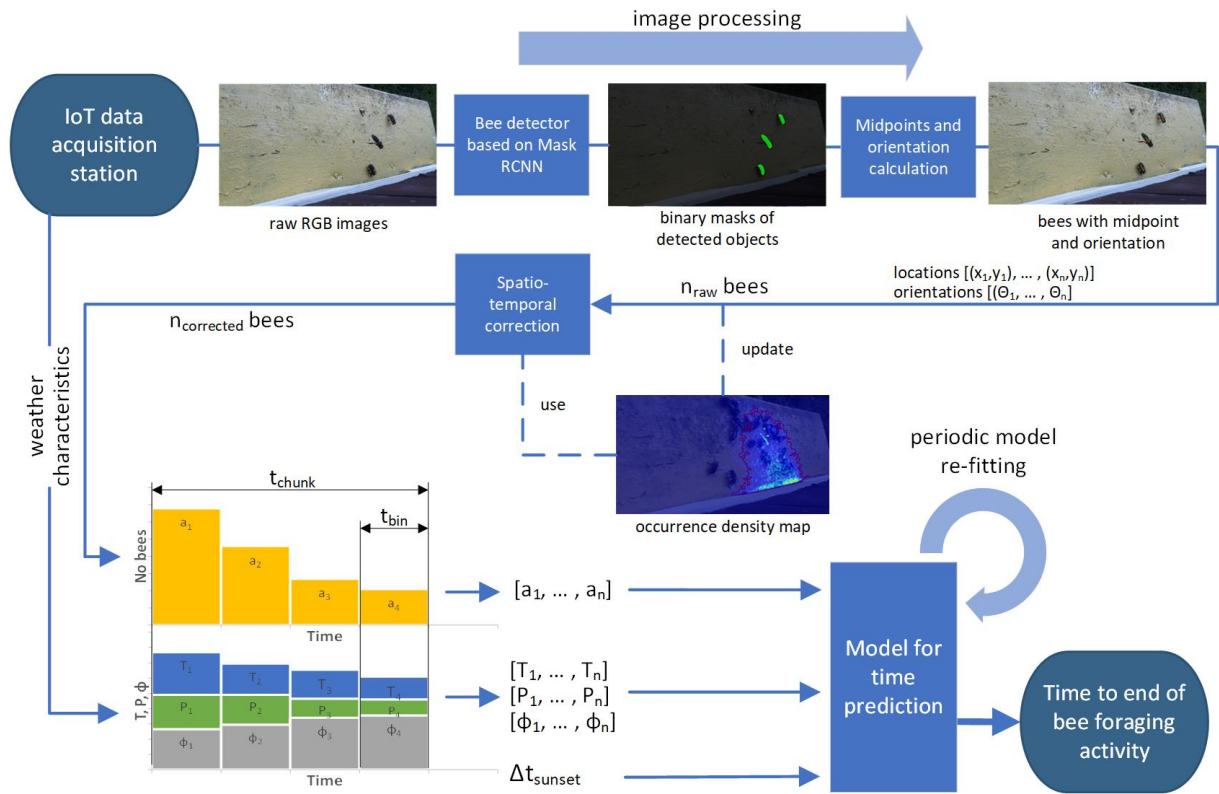


Figure 2.4: The proposed solution for predicting the time remaining to the end of the bees' daily foraging activity [A2].

was used to determine the training sets for the single models in the ensemble. Based on the standard deviation of the prediction in the model ensemble, the uncertainty of the prediction was determined, which was used in further steps to detect the domain shift effect. The domains proposed in the article [A4] were used for the study. Finally, the proposed method for detecting the domain shift effect based on averaged values of prediction uncertainty (from 10 samples) was characterized by an accuracy of F1-score  $> 0.94$ .

In article [A7] I addressed the domain adaptation problem in the context of the re-identification of mealworm beetles. The observed domain shift effect was related to the change in the character of the recorded images between the training and testing stages of the proposed phenotyping procedure. The proposed method involved re-training the re-identification model on a set containing samples from the training stage (with true labels) and samples from the testing stage (with pseudo labels determined by the prediction of the initial re-identification model and the confidence score threshold). The proposed domain adaptation method for the re-identification model made it possible to increase precision-1 from 0.807 to 0.853.



**ALGORITHM 3**


---

Method for adapting instance segmentation model to new domain with synthetic images

---

**Input:** set of labeled real images from source domain  $S_{real} = \{s_1, s_2, \dots, s_n\}$

set of unlabeled real images from target domain  $T_{real} = \{t_1, t_2, \dots, t_m\}$

confidence score threshold  $cs_{thresh}$

**Output:** instance segmentation model after 2. stage

▷ 1. stage of domain adaptation procedure

**extract** objects from images in  $S_{real}$  determining pool of objects from source-domain  $P_{source}$

**augment** objects from  $P_{source}$  determining  $P_{source}^*$  (object pool with augmented objects)

**generate** synthetic images using  $P_{source}^*$  determining set of synthetic images  $S_{syn}$

**train** model on  $S_{real}$  and  $S_{syn}$

▷ 2. stage of domain adaptation procedure

**predict** labels (masks) for images from  $T_{real}$  using model from stage 1. and  $cs_{thresh}$

**extract** objects from images in  $T_{real}$  using predicted masks determining pool of objects from target-domain  $P_{target}$

**filter out** objects from  $P_{target}$  using knowledge-based rules obtaining  $P_{target}^*$

**augment** objects from  $P_{target}^*$  determining  $P_{target}^{**}$  (object pool with augmented objects)

**generate** synthetic images using  $P_{target}^{**}$  determining set of synthetic images  $T_{syn}$

**train** model on  $S_{real}$ ,  $S_{syn}$  and  $T_{syn}$

---

[O3] Development of method enabling reduction of complexity (inference time) of machine learning methods for phenotyping insect biosystems, involving knowledge transfer and end-to-end model

**Relevant publications:** A3, A6

**Description:**

In article [A3] I proposed a method for determining size indices of mealworm larvae (lower quartile, median, upper quartile of larval width) using a regression convolutional neural network (RegCNN). The developed solution addressed the problem of a time-consuming reference approach based on multistage processing using classical computer vision methods. The proposed approach made it possible to reduce the inference time per breeding box from 10.9 s to 0.3 s while maintaining a relatively small error in determining the size indices of larvae ( $R^2 = 0.870$ ). In developing RegCNN, the standard procedure of manual labelling of samples was omitted through synthetic data generation (described in more detail for the research objective [O1]) and knowledge transfer. The knowledge transfer consisted of training RegCNN using the outputs obtained from the multistage processing using classical computer vision methods. It was confirmed that during knowledge transfer, the loss of accuracy was relatively small and acceptable from the point of view of the considered problem (the reference value of  $R^2$  was 0.927). The knowledge transfer method with the proposed pseudo-targets correction is shown in Figure 2.5. In the approach presented in Figure 2.5, the pseudo-targets correction method should be indicated as the main novelty. The correction was motivated by the different proportions of correct detections for objects of different sizes in images

representing dense scenes.

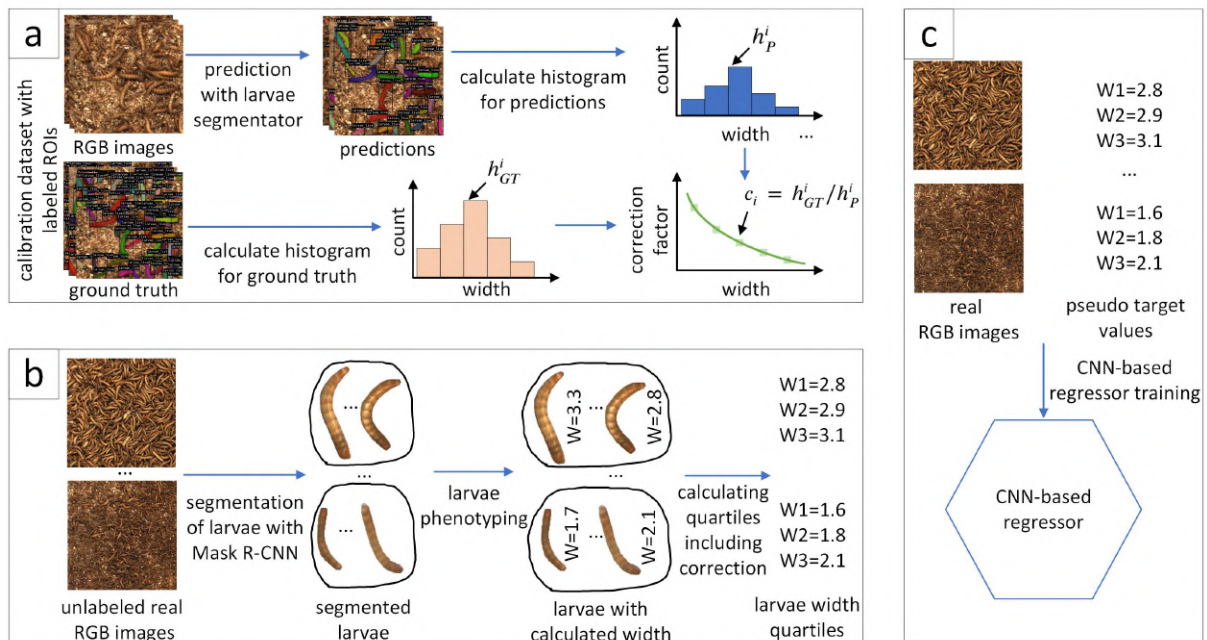


Figure 2.5: The knowledge transfer method: (a) correction factor determination for width quartiles calculation, (b) multistage phenotyping for selected samples, and (c) training of a regression convolutional neural network using knowledge transfer [A3].

In article [A6] I developed an end-to-end model for calculating multiple indices that characterize the current state of mealworm rearing (counts of specific growth stages, anomalies, image coverage coefficients of chitinous moults and feed, and size indices). The proposed solution made it possible to replace multiple modules related to a specific task and a separate model (the approach used in the article [A1]) with a single condensed architecture, which significantly reduced the complexity of the solution and, at the same time, the inference time and ease of maintaining the model during production. Each head, based on embeddings extracted from a specific layer of the YOLOv8n model, predicted the value of the chosen indicator using a previously trained classical regression machine learning model (such as linear regression, and gradient boosting regression). The described multi-task end-to-end model is shown in Figure 2.6. The distinguishing characteristic (novelty) of the proposed end-to-end architecture is its versatility and ease of extension, as it can be applied to any problem that can be reduced to a finite number of object detection and regression problems. Another highlight is that the proposed architecture allows separate training of individual heads on problem-oriented datasets, as described in more detail in the description to the research objective [O1].

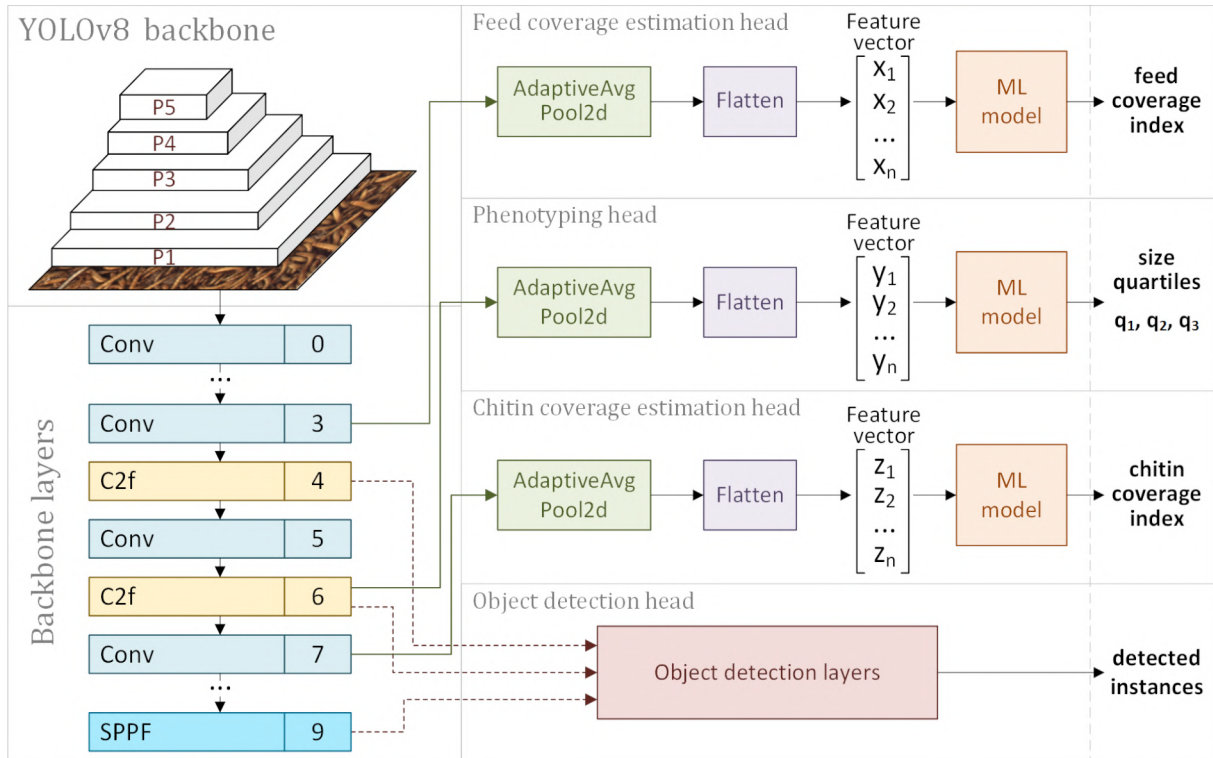


Figure 2.6: The end-to-end architecture with proposed additional heads: feed coverage estimation head, chitin coverage estimation head, and phenotyping head [A6].

[O4] Development of method enabling the incorporation of domain knowledge (a priori) in the development, maintenance, and inference of machine learning methods for phenotyping insect biosystems

**Relevant publications:** A2, A4, A5

**Description:**

The issue of incorporating domain knowledge in the articles [A2, A4, A5] was addressed in the context of various problems, namely improving the development, maintenance and inference of machine learning models.

In paper [A5] I used domain knowledge to improve the inference accuracy of pest detection models. With the knowledge of the pest's higher mobility compared to the mealworm, a spatio-temporal masking method was proposed. Spatio-temporal masking was based on isolating an area in the image where the probability of finding the pest was highest. The determined activity maps after thresholding were used to determine the mask. The activity maps were calculated using the Gunnar Farneback optical flow algorithm. It was shown that under conditions of low/moderate pest infestation in the rearing box (the most common conditions), the proposed approach increased the accuracy of pest detection F1-score from 61.7 to 66.6 (with the number of samples in the training/validation set of 631). The improvement was also observed with fewer samples in the training/validation set.

In article [A4], I used domain knowledge in the second stage of the proposed method for domain adaptation for the detection model of three states of the mealworm: live larva, dead larva and pupa. Using assumptions based on domain knowledge, filtering objects in the object pool for generating synthetic images was done. The following assumptions were used: (1) the live larva is the majority class (the number of objects from the live larva class in the images is the highest), (2) objects from the live larva and dead larva classes are longer than objects from the pupa class (length is defined here as the length of the longer side of the bounding box), (3) objects from the pupa class have the highest averaged pixel intensity among the considered classes, and (4) objects from the dead larva class have the lowest averaged pixel intensity among the considered classes. The proposed filtering strategy made it possible to remove many falsely classified objects from the target domain from the pool of objects for generating synthetic images. The second stage of the method for domain adaptation (including knowledge-based filtering) made it possible to increase the accuracy of the detection model from AP50 62.9 to 71.8 (results for inference in the target domain).

In article [A2] I used domain knowledge in a spatio-temporal correction method to more accurately determine pseudo-targets for periodic model re-fitting. The observed behavioural patterns of bees (ventilating the hive, cleaning the hive by detecting immobile dead individuals) at the entrance to the hive made it very difficult to determine the time of the end of the bees' daily flight. The correction introduced was based on assigning smaller weights to bees whose position did not change between recorded frames (this pattern was related to the described behavioural patterns) when counting. In the description for the research objective [O2], quantitative improvement in model performance was shown when using spatio-temporal correction and periodic model re-fitting methods.

[O5] Development of method enabling phenotyping insect biosystems at the level of individuals (rather than population), involving re-identification and detection of behavioural patterns

**Relevant publications:** A7

**Description:**

In paper [A7] I proposed a method for the re-identification of *Tenebrio molitor* beetles based on fine-tuned feature extractors pre-trained on ImageNet and metric learning. The pattern of mating behaviour was detected using the YOLOv8 object detection model. The article [A7] confirmed that re-identification of *Tenebrio molitor* beetles is possible with high accuracy based only on the appearance of the beetles' abdomen (without additional markers). Precision-1 metric values of 0.853 (after applying domain adaptation) were achieved with 80 analyzed individuals. The hard validation was based on physical markers placed on the beetles' heads, which ensured the experiment's reliability. After ablation studies, it was shown that features related to colour mainly enabled re-identification. Texture and shape features were not sufficient to perform re-identification with high accuracy. The paper [A7] also showed that detecting behavioural patterns, i.e., mating is possible with relatively high accuracy with a small set of labelled samples. AP50 values for detection of 0.835 were achieved (after

using additional synthetic images) with a total number of labelled samples in the dataset of 173. Other relevant parts related to developing models for the re-identification of individuals described in the article [A7] are included in the research objectives O1 and O2 descriptions.



# CHAPTER 3

## Conclusion and Future Work

---

The dissertation proposed many methods for rapid development, efficient maintenance, adaptation, and reducing the complexity of machine learning models for phenotyping insect biosystems. Moreover, the proposed techniques for integrating domain knowledge and re-identification underscored the feasibility of developing dedicated models for phenotyping insect biosystems. The generated synthetic images confirmed their usefulness for rapid model development and were also used in the original domain adaptation method. Methods based on semi-supervised learning enabled the effective use of unlabelled samples in model training and maintenance through pseudo-label-based self-training. The importance of semi-supervised learning was particularly significant in the problem of weakly represented datasets. Knowledge transfer, enabling end-to-end model training on the predictions of methods based on multistage image processing, was very useful in reducing the complexity and inference time of the developed solutions. The developed re-identification techniques and the individual phenotyping procedure provide new insights into the problem of insect biosystem analysis, opening up a wide area for research.

In summary, the research carried out, as part of the dissertation, confirmed the research hypothesis set out, i.e. machine learning methods using synthetic images, semi-supervised learning, knowledge transfer and end-to-end architectures enable the development of dedicated models for phenotyping insect biosystems that are more efficient, easier to develop and maintain and characterized by shorter inference times than currently used machine learning methods. All set research objectives (O1-O5) were met and defined research gaps (RG1 - RG6) were filled.

The most interesting direction for future work is to focus on individualized phenotyping with the detection of insect behavioural patterns based on interactions. Approaches based on graph neural networks may allow the description of complex interactions between individuals and their behaviour in the studied biosystems, making it possible to determine their welfare. Also, the problem of extracting knowledge about insect (animal) biosystems from large amounts of unlabeled data with the development of semi-supervised and unsupervised methods is still open.





# CHAPTER 4

## Publications

---

This chapter presents the articles that were included in the dissertation. Six articles were published or accepted for publication. One article is under review. The PhD candidate's contribution to each article was characterized in detail.

### 4.1 Multipurpose monitoring system for edible insect breeding based on machine learning

**Authors:** Paweł Majewski, Piotr Zapotoczny, Piotr Lampa, Robert Burduk, and Jacek Reiner

**Publication status:** published

**Type of publication:** journal paper

**Journal/Conference:** Scientific Reports (IF=4.6)

**MEiN points:** 140

**Lead Author:** Yes

**Corresponding Author:** Yes

**Percentage contribution:** 60%

**CRedit:** Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing–original draft preparation, Writing–review and editing



OPEN

# Multipurpose monitoring system for edible insect breeding based on machine learning

Paweł Majewski<sup>1✉</sup>, Piotr Zapotoczny<sup>2</sup>, Piotr Lampa<sup>3</sup>, Robert Burduk<sup>1</sup> & Jacek Reiner<sup>3</sup>

The *Tenebrio molitor* has become the first insect added to the catalogue of novel foods by the European Food Safety Authority due to its rich nutritional value and the low carbon footprint produced during its breeding. The large scale of *Tenebrio molitor* breeding makes automation of the process, which is supported by a monitoring system, essential. Present research involves the development of a 3-module system for monitoring *Tenebrio molitor* breeding. The instance segmentation module (ISM) detected individual growth stages (larvae, pupae, beetles) of *Tenebrio molitor*, and also identified anomalies: dead larvae and pests. The semantic segmentation module (SSM) extracted feed, chitin, and frass from the obtained image. The larvae phenotyping module (LPM) calculated features for both individual larvae (length, curvature, mass, division into segments, and their classification) and the whole population (length distribution). The modules were developed using machine learning models (Mask R-CNN, U-Net, LDA), and were validated on different samples of real data. Synthetic image generation using a collection of labelled objects was used, which significantly reduced the development time of the models and reduced the problems of dense scenes and the imbalance of the considered classes. The obtained results (average  $F1 > 0.88$  for ISM and average  $F1 > 0.95$  for SSM) confirm the great potential of the proposed system.

The current problems of feeding an ever-increasing human population involve meeting the demand for animal protein without the environmental costs associated with animal husbandry. Preference is given to livestock systems that use less water, minimise space and reduce greenhouse gas emissions. The United Nations (UN) predicts that human protein consumption will reach 39 grams per day in 2030, and 57 grams in 2050<sup>1</sup>. The solution to this problem may be industrial insect breeding with minimised human labour and high stocking rates per unit building area. This is important, because in recent years, according to the recommendations of good husbandry practices, there has been an aim to reduce the stocking density per 1m<sup>2</sup> of building area of the main livestock species such as cattle, pigs and poultry. Moreover, we have also observed huge problems with African swine fever (ASF) and Avian influenza (AI), causing many livestock buildings to close with no idea of how to then use them. One alternative for their reuse could be intensive breeding of insects for food and feed<sup>2</sup>. According to the International Platform of Insects for Food and Feed (IPIFF), within the next 10 years the insect sector will become an integral part of the European agri-food chain. It is forecast that 1 in 10 fish consumed in the European Union (EU) will come from fish farms that use insect protein in their feed, 1 in 4 eggs consumed in Europe will come from insect-fed laying hens, 1 in 5 servings of chicken meat will come from insect-fed broilers, and 1 in 100 servings of pork will come from insect-fed pigs.

Insects are the most numerous group of known animal species, and they are the most important element of the ecosystem<sup>2</sup>. They are a valuable source of protein for many animal species and for people living in Africa, Asia or South America. Of the millions of insect species, more than 2000 are recognised as being edible. In Europe, there is no tradition of eating insects as a protein substitute. For now, most are bred only as protein and fat supplements to feed other animals. This is possible because the EU has approved insect protein for the production of feed for fish, poultry and pigs. Additionally, in 2021, after many studies, the European Food Safety Authority showed that *Tenebrio molitor* larvae are a rich source of protein, fat and fiber, and included it in the catalogue of novel foods. Thus, whole or powdered dried larvae can be an ingredient in pasta, cookies, and other food products. However, for such breeding to be profitable, industrial production technologies for selected insect

<sup>1</sup>Faculty of Information and Communication Technology, Wrocław University of Science and Technology, Wrocław, Poland. <sup>2</sup>Department of Systems Engineering, University of Warmia and Mazury in Olsztyn, Olsztyn, Poland. <sup>3</sup>Faculty of Mechanical Engineering, Wrocław University of Science and Technology, Wrocław, Poland. ✉email: pawel.majewski@pwr.edu.pl

species, such as *Tenebrio molitor*, must be developed in order to provide a standardised and cost-competitive product to the market.

*Tenebrio molitor* is a beetle from the Darkling beetles (Tenebrionidae) family. Adults have an elongated body measuring 12–18 mm in length. In the life cycle of *Tenebrio molitor*, the following can be distinguished: eggs, larvae, pupae and beetles. The larvae transform into a pupae in 45–60 days after 7–9 moult cycles, and the pupae transform into beetles after 5 next days<sup>3</sup>.

Industrial livestock production technology involves fully automating almost all animal handling operations through robots supervised by vision systems and a set of sensors associated with a production management module. If industrial breeding of *Tenebrio molitor* is to be profitable, individual breeding processes should also be automated. Its breeding systems currently involve keeping a certain number of larvae in boxes that are stacked on racks at several or more levels. Most of the work performed is manual and labour intensive. Today, with the exception of a few very large operators that are capable of producing several thousand tons of insects, the vast majority of insect farms are startups and small or medium-sized companies with little or no automation in their production systems. Keeping in mind the recommendation for breeding<sup>4</sup>, the activities in the production of *Tenebrio molitor* are: (i) feeding, (ii) wetting of larvae, (iii) sorting of larvae into size classes, (iv) harvesting of chitinous moult, (v) final harvesting of larvae and the separation of them from impurities. In order to control the ongoing effects of breeding, it is necessary to measure: (1) biomass gains, (2) the amount of chitinous moult, (3) the amount of dead larvae, (4) the amount of consumed feed, (5) the amount of possible pests (*Alphitobius diaperinus*), and (6) the number of individuals after transformation to pupae or beetles. In view of the requirements, fully automating production is not something that is easy. The basis of farm automation can be a vision system based on RGB cameras, or cameras outside the visible range (UV, IR). However, the problem is not with the hardware, but with the software. While the availability of cameras is very high, there is a lack of information in literature on the algorithms that can identify even the basic parameters of *Tenebrio molitor* breeding. This problem is difficult to solve because the objects to be identified overlap, and the colour or texture of each instance is very similar to each other.

Image analysis methods have more and more applications in precision agriculture. They are commonly used to assess the quality of raw materials and food products<sup>5</sup>. There is also research using vision systems and soil worms to assess drug effects. Digital fluorescence images of *Caenorhabditis elegans* worms were captured with a CCD camera, and the lymph flow through the worms' bodies was determined based on the developed algorithms (Migliozzi et al., 2019)<sup>6</sup>. Tao et al.<sup>7</sup> presented the results of identifying the sex of silkworm pupae using vision systems based on hyperspectral cameras. They used the successive projections algorithm (SPA) for variable selection<sup>8</sup>, gray-level co-occurrence matrix (GLCM)<sup>9</sup> analysis, and support vector machines (SVM)<sup>10</sup> and radial basis function neural network (RBF-NN) models to achieve more than 98% accuracy in identifying the sex of silkworm pupae. Similar results were obtained by Sumriddetchkajorn et al.<sup>11</sup> except that they obtained images of silkworm pupae by illuminating the cocoons with light from diodes and then capturing the images with a CCD camera. A combination of near-infrared hyperspectral imaging, convolutional neural networks (CNN), and a capsule network<sup>12</sup> allowed for the identification of the storage pest (khapra beetle, *Trogoderma granarium* Everts) with over 90 percent accuracy (Agarwal et al.)<sup>13</sup>. A study on determining the developmental stage of pupals using vision systems was conducted by Sasha et al.<sup>14</sup> for two species of blowfly (Diptera: Calliphoridae).

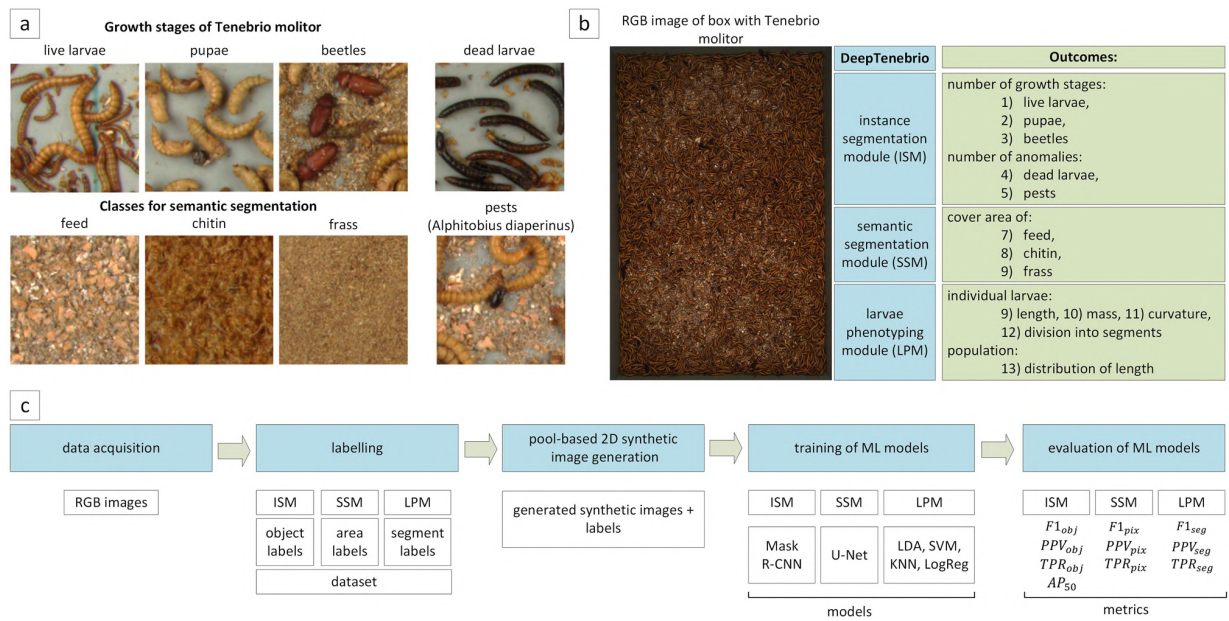
Unfortunately, there are only a few articles on the use of image analysis to automate *Tenebrio molitor* production. Companies with solutions, such as Dilepix, offer off-the-shelf systems, but do not provide details of their solutions. Kröncke et al.<sup>15</sup> presented a system for automating the production of *Tenebrio molitor*. They developed a pneumatic system for separating larvae from impurities. They also proposed a method to evaluate the health and developmental status of larvae using a vision system. For this purpose, they classified image fragments into three classes: good segments, bad segments, and artifacts with the use of a multi-layer perceptron neural network (MLP-NN), which achieved an accuracy of 95.4%.

Due to a lack of sufficient knowledge on the development of complex systems for the automatic production and industrial control of *Tenebrio molitor*, the authors undertook to develop such a system. Its key element is a vision system, the tasks of which include the automatic identification of individual instances and the calculation of production parameters, which are the basis for the control of the entire farm. The main achievements of our research are: (1) a multipurpose monitoring system for edible insect breeding based on machine learning, (2) a novel non-invasive method for calculating the mass of *Tenebrio molitor* larvae based on images, (3) a novel method for estimating the size distribution of objects in dense scenes, (4) an original method for developing models for multiclass instance and semantic segmentation based on synthetic image generation and a partially automated labelling process.

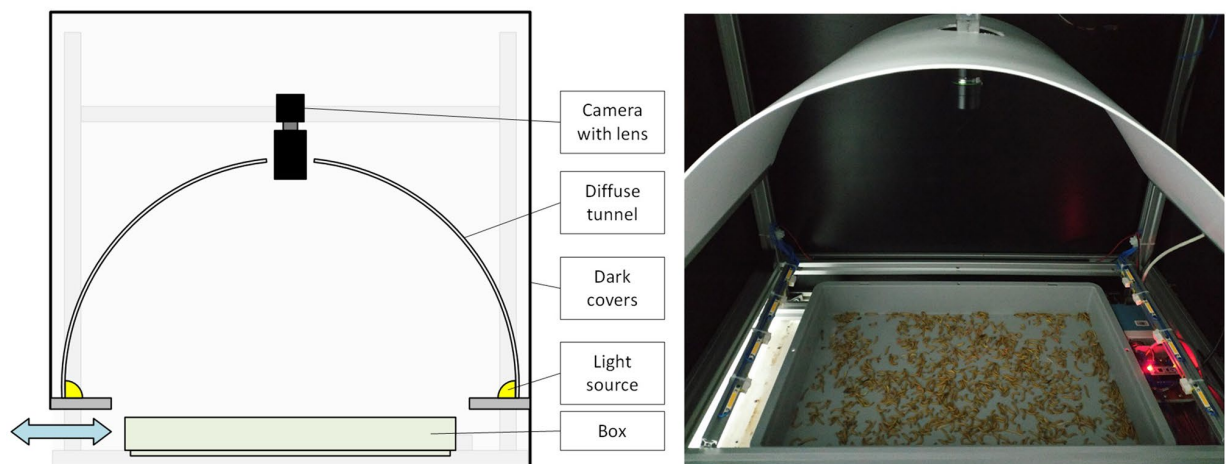
## Methods

This chapter describes the successive steps of the iterative development of machine learning models, from defining problems and system concept to the evaluation of the proposed methods.

**Problem definition and system concept.** We divided the addressed problems into 3 groups: those requiring instance segmentation, those requiring semantic segmentation, or those related to larvae phenotyping. The first group involved the detection and segmentation of the growth stages of *Tenebrio molitor* (live larvae, pupae, beetles), and also anomalies in the form of dead larvae and pests (*Alphitobius diaperinus*). The instance segmentation module (ISM) determined the number of objects belonging to each class, their location, and an extracted binary mask for further phenotyping of individual instances. The second semantic segmentation module (SSM) extracted areas that represent the densities of the feed, chitin and frass from the image, and then calculated the percentage of these areas in the whole image. The third module was related to larvae phenotyping



**Figure 1.** Presentation of the addressed problem: (a) the defined object classes used in the study, (b) the concept of the proposed 3-module DeepTenebrio system, (c) the next steps in developing the machine learning models for the proposed modules.



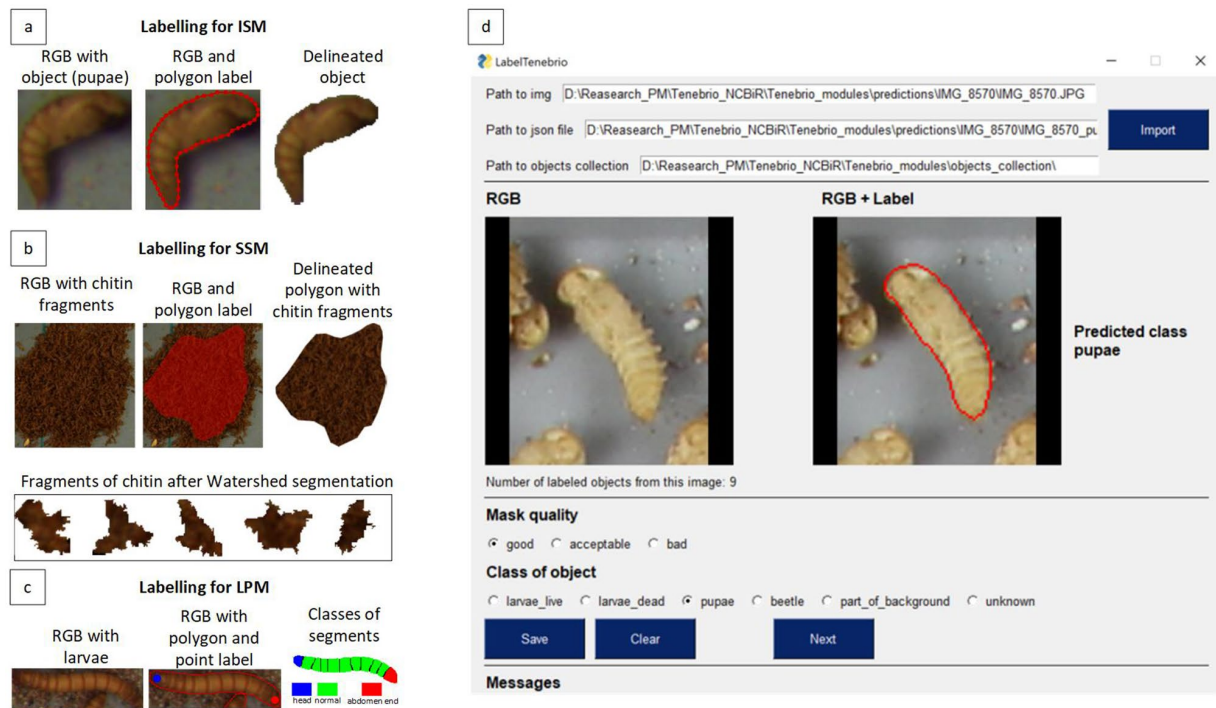
**Figure 2.** Data acquisition station.

(LPM) at the level of both individual larvae (calculation of length, curvature, mass, division into segments, and their classification) and the whole population (determination of length distribution). The defined object classes used in the study (a) and the concept of the proposed 3-module DeepTenebrio system (b) are shown in Fig. 1.

Figure 1 also shows the next steps in developing the machine learning models for the proposed modules (c), which are characterised in the following sections.

**Data acquisition.** The place for breeding *Tenebrio molitor* were boxes placed on the shelves of racks. Selected boxes with *Tenebrio molitor* were taken off the shelves and put into the data acquisition station. Its schematic diagram, with a real photo, is shown in Fig. 2.

The station allowed high-resolution RGB images in manual and automatic modes to be collected. The Phoenix PHX120S-CC camera (LUCID Vision Labs, Canada) with a resolution of 4096 x 3000 pixels was selected for image acquisition. The boxes with *Tenebrio molitor* were illuminated with a neutral white light (colour temperature 4000K) that was scattered in a diffuse tunnel. To reduce insect stress, the illuminators were only triggered for a short time for the duration of camera exposure. The covers isolated the image acquisition area from external factors. In total, 120 raw images of boxes with *Tenebrio Molitor* under breeding conditions were collected as a basis for labeling and developing the proposed modules. The selected populations in boxes differed



**Figure 3.** Methods and tools used for the labelling: (a) object labelling for the ISM, (b) area labelling for the SSM, (c) larva segment labelling for the LPM, (d) LabelTenebrioApp for improving the instance labelling process.

in the growth stage of individuals, presence of anomalies, amount of uneaten food and chitin. Data were collected at Tenebria (Lubawa, Poland).

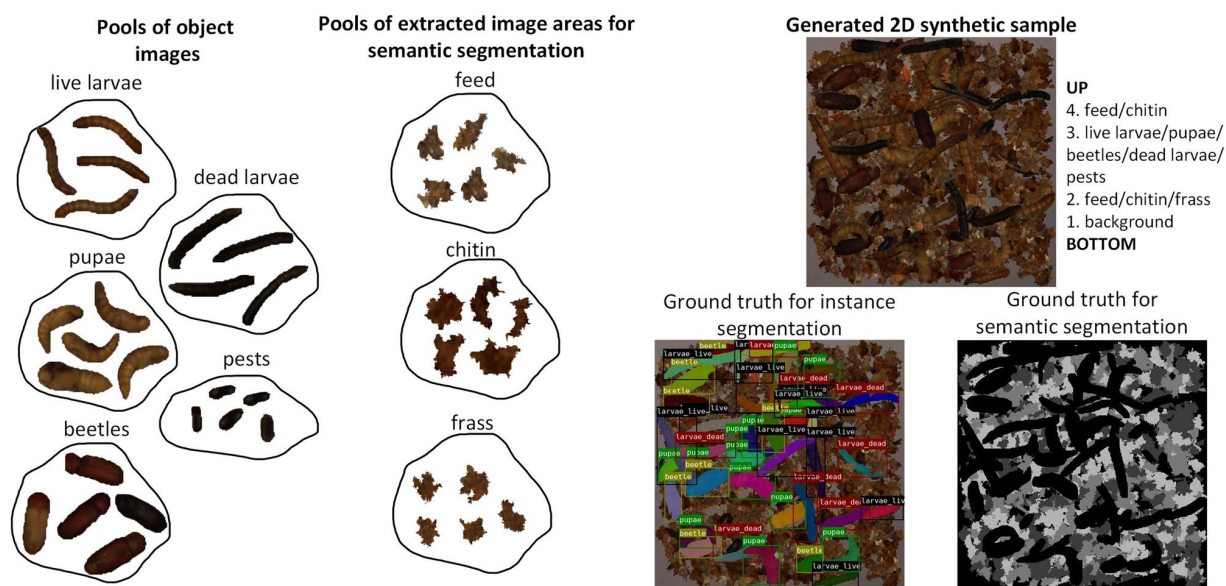
**Labelling.** Labelling is an integral part of developing machine learning models, and allows the transfer of annotator knowledge to the model through its supervised training. This section describes the adopted data labelling strategy for training the models for the following modules: ISM, SSM, and LPM, as shown in Fig. 1.

*First stage of labelling.* The first stage of labelling consisted of manual annotation of the images. For each proposed module: ISM, SSM, and LPM, the forms of annotation were different, and are shown in Fig. 3.

In the case of the ISM (a in Fig. 3), the labelling consisted of delineating the boundaries of consecutive objects with a polygon-type label in order to obtain an object mask, to extract the object from the image, and then to add it to the object pool. For the SSM (b in Fig. 3), areas representing only one class, e.g. chitin, were marked, and a polygon-type label was also obtained. These areas were then divided into smaller fragments using the Watershed algorithm, which were then added to the pools associated with the classes for semantic segmentation. Samples for larval segment classification for the LPM (c in Fig. 3) were obtained as follows. On an extracted live larva mask, two point labels were placed at the two ends in order to denote the head and abdomen end. The end segments were assigned to the head or abdomen end class depending on the annotator label, and the rest of the segments obtained the “normal” label (normal abdomen segments). The algorithm for dividing the larvae into segments was unsupervised, and is described in “Larvae Phenotyping” section.

*Second stage of labelling.* Obtaining an efficient and robust machine learning model requires iterative model development. This is not only related to retraining the models on enlarged datasets, but also to using previous models (so-called weak models) in order to achieve improved labelling. This makes it possible to quickly find the most difficult samples for inference, as well as to annotate them manually, which is the assumption of active learning<sup>16</sup>. Labelling samples for the instance segmentation model was very time-consuming. For this reason, an improved labelling process based on LabelTenebrioApp (d in Fig. 3) was proposed. At first, the annotator selected an image, together with previously obtained predictions, using the model with the best results so far. The annotation process itself involves a quick evaluation of the received predictions in terms of mask quality, and then predicted object class using a point-click technique. Objects with masks of good/acceptable quality and a true class label were added to the object pools without the need, as is the case with classical labelling, to draw a polygon. The most difficult cases that were not identified by the model were then manually labelled by the annotator.





**Figure 4.** Pool-based 2D synthetic image generation method.

Dataset type	No. of objects (ISM)					No. of polygons (SSM)			No. of segments (LPM)		
	Live larvae	Pupae	Beetles	Dead larvae	Pests	Feed	Chitin	Frass	Head	Normal	Abdomen end
Training	1026	1550	242	809	133	22	6	11	222	1879	225
Test	346	199	68	140	61	42	85	13	36	287	36

**Table 1.** Number of objects in the training and test datasets.

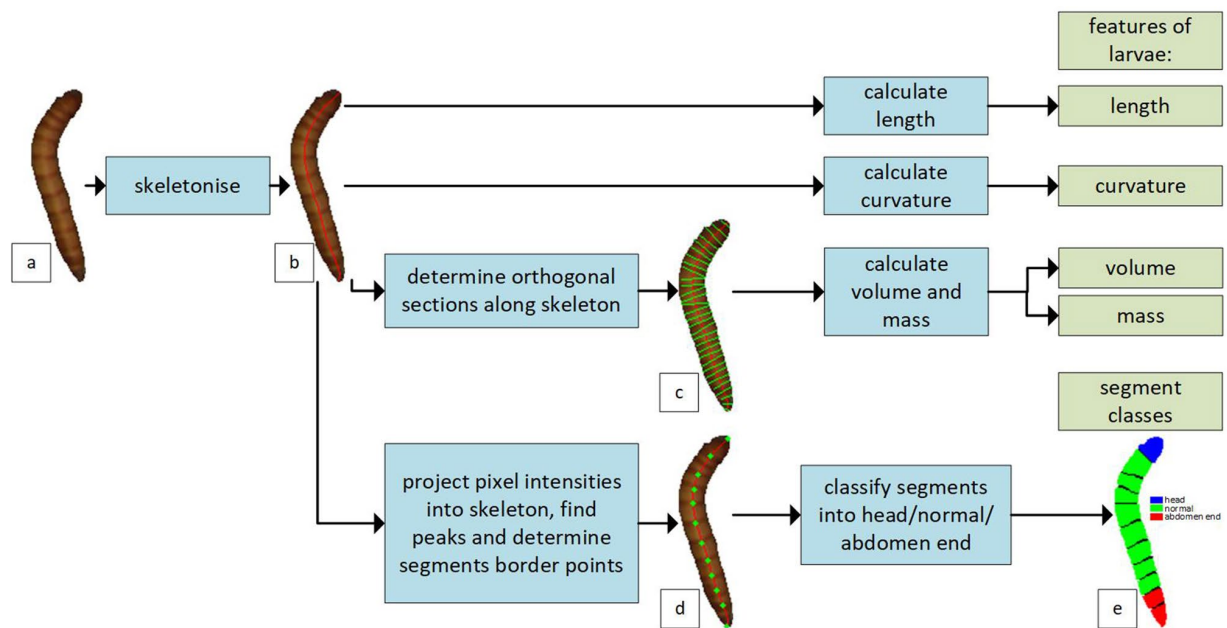
**Pool-based 2D synthetic image generation.** Developing machine learning models in the classical way, i.e., based on the completely manual annotation of random samples, is inefficient. This is especially noticeable for segmentation problems, when the label is a pixel annotation that requires a lot of effort from the annotator. For the undertaken issues, another problem is the high density of objects, their overlap, and their similarity, which increases uncertainty during labelling.

Considering these limitations, a semi-automatic method for generating 2D synthetic samples was proposed, which significantly reduced labelling time and uncertainty, and increased flexibility for iterative model development. The method is based on randomly placing elements in the form of images from the pools on the background image<sup>17</sup>. The extraction of items for the pools is described in "Labelling" section. In order to obtain a high similarity between the synthetic and real data, the elements were placed in a specific order. The first were fragments of feed, chitin, and frass, which are often found at the bottom of the box. Next, images of objects such as live larvae, pupae, beetles, dead larvae, and pests were placed in the image. Feed and chitin fragments were placed in the foreground, which is due to the fact that there could be possible feed residues after feeding, as well as moulting during larval growth. After the item placement procedure, a label was automatically generated in order to train the instance and semantic segmentation models without additional user supervision. A diagram of the proposed pool-based synthetic data generation method is shown in Fig. 4.

**Dataset.** The labelling strategy described in "Labelling" section was applied to create the training datasets. A summary of the number of samples collected in this way is shown in Table 1.

The numbers shown in Table 1 represent the number of objects for the ISM, the number of polygons for the SSM, and the number of segments for the LPM, respectively. The test datasets were prepared completely independently. In this case, the samples were labelled manually by an expert on varied images of boxes with *Tenebrio molitor*. The labelling process was performed similarly to the first labelling stage described in "Labelling" section except that the steps of preparing and adding items to the group of objects were omitted, e.g. areas were not divided into smaller fragments in the case of semantic segmentation. A summary of the number of test samples is given in Table 1. The numbers have analogous meanings to the training datasets.

**Instance segmentation with mask R-CNN.** The Mask R-CNN<sup>18</sup> model was used for the instance segmentation of objects from the classes: live larvae, pupae, beetles, dead larvae, pests. The Mask R-CNN complements the Faster R-CNN<sup>19</sup>, and has a part that is responsible for generating a mask for each detected object. The functioning of both the Mask R-CNN and Faster R-CNN is based on the determination of the region of interest



**Figure 5.** Larvae phenotyping schema: (a) raw image of the larva, (b) skeleton of the larva after skeletonisation, (c) orthogonal sections to the skeleton for calculating the volume of the larva, (d) segment boundary points marked on the skeleton, (e) segment classification.

by the Region Proposal Network (RPN) that is based on the feature map, which is the output of the convolutional neural network with the selected architecture. Once the region of interest sizes are unified, classification and boundary box regression using Fully Connected Layers is performed. The Mask R-CNN model additionally makes a mask prediction at this point. The loss minimised during training takes into account the accuracy of the described three model predictions, namely classification, bounding box regression, and mask extraction. In this study, ResNet-101<sup>20</sup> was used as the feature map extractor, which is a common choice of researchers for similar problems<sup>21</sup>. The following hyperparameters were adopted for Mask-RCNN training: optimizer SGD, learning rate  $2.5 \times 10^{-4}$ , iterations 10,000, weights mask\_rcnn\_R\_101\_FPN\_3x. The research used the Mask R-CNN implementation from the Python library detectron2<sup>22</sup>.

**Semantic segmentation with U-net.** The U-Net model<sup>23</sup> was used for the semantic segmentation of the feed, chitin and frass areas from the images. U-Net has an autoencoder structure that consists of three main parts: an encoder, and a decoder with an identical structure and a bottleneck. The autoencoders encode information in the bottleneck and then decode it, resulting in the extraction of only the most important patterns from the data, as well as the reduction of noise. The dice loss minimised during training the model is based on comparing the model output with the ground truth. The values in the model output are scaled to the probabilities for the given classes using the softmax activation function. The final performance of the model is influenced by the choice of encoder and decoder architecture. For the issues undertaken, the EfficientNet-B0<sup>24</sup> backbone pre-trained on ImageNet<sup>25</sup> was chosen due to its efficiency and relatively small size. The following hyperparameters were adopted for U-Net training: optimizer Adam, learning rate  $10^{-4}$ , epochs 40. The study used the U-Net implementation from the Python library segmentation\_models<sup>26</sup>.

**Larvae phenotyping.** The LPM allows the determination of the basic characteristics of individual larvae (length, curvature, mass, division into segments, and their classification), and also the distribution of larval length for the whole population. This section describes the methods used in the LPM. The phenotyping scheme for single larvae is shown in Fig. 5.

**Length calculation.** Determining the length of larvae is not an obvious task due to the need to strictly define this dimension. In this research, the larval length was assumed to be the length of a curve going through the center of the larva along their largest dimension. To determine the described curve, the skeletonisation algorithm<sup>27</sup> was used, along with a correction for boundary conditions, which involved drawing additional pixels at the ends of the curve while taking into account the local orientation of the skeleton and mask boundaries. The skeleton for an example larva is shown in image b in Fig. 5. By having the skeleton coordinates, the length was calculated from the following formula:

$$L = k \sum_{i=1}^{n-1} l(S_{i+1}, S_i) = k \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (1)$$

where  $l(S_{i+1}, S_i)$  - the Euclidean distance between consecutive points of the skeleton  $S_{i+1}(x_{i+1}, y_{i+1})$  and  $S_i(x_i, y_i)$ ,  $k$  - the constant enabling the conversion from pixels to millimetres,  $n$  - the number of points in the skeleton.

**Volume and mass calculation.** When calculating the volume of larvae, it was assumed that it can be approximated by the total volume of a finite number of cylinders, the height  $l_i$  of which is equal to the length of the defined skeleton section, and the diameter  $d_i$  is equal to the length of the orthogonal section to the defined skeleton section that is contained within the binary mask, as shown in image c in Fig. 5. Once these quantities are determined, the value of the larvae volume can be calculated from the formula:

$$V = k^3 c \sum_{i=1}^{n-1} \frac{\pi}{4} d_i^2 l_i \quad (2)$$

where  $l_i$  - the Euclidean distance between consecutive points of the skeleton,  $d_i$  - the length of the section orthogonal to the considered section on the skeleton,  $k$  - the constant enabling the conversion from pixels to millimetres,  $n$  - the number of points in skeleton,  $c$  - the correction coefficient.

The use of correction factor  $c$  in the estimation of larval volume is due to the differences occurring between the real shape of the larvae and the ideal shape assumed in the study, which is especially affected by the flattening of the thorax near the head, and the lack of volume at the joints of the larval segments. Empirically, the value of the correction factor was determined to be  $c = 0.58$ . The value of the constant  $k$  should always be determined individually during calibration using the length calibration standard (the value of the constant  $k$  depends on the resolution of the camera and the dimensions of the area of interest, e.g. box). A constant  $k$  equal to 0.153 [mm/pixel] was used in this study. For larvae mass calculations, the empirically determined density of mature *Tenebrio molitor* larvae equal to  $\rho = 1.31 \pm 0.25 \frac{\text{g}}{\text{cm}^3}$  was used. The density was measured using a HumiPyc gas pycnometer.

**Curvature calculation.** Another calculated parameter was curvature. By knowing the coordinates of the skeleton, the value of curvature at a certain point  $S(x, y)$  can be calculated from the formula:

$$\kappa = \frac{|x'y'' - y'x''|}{[(x')^2 + (y')^2]^{\frac{3}{2}}} \quad (3)$$

where  $x$  and  $y$  are coordinates of the skeleton points  $S(x, y)$ , and  $x', y', x'', y''$  are 1. and 2. order derivatives for a given coordinate.

The curvature of the larvae was calculated for the averaged coordinates of the skeleton points in the defined intervals with specific lengths. The final referenced curvature value is equal to the average curvature value at the specified points.

**Division into segments and their classification.** The larvae of *Tenebrio molitor* were composed of segments. The segments contained in the middle were similar and were characterised by a segment ending in the form of a dark band orthogonal to the skeleton. This pattern was used in the unsupervised division of the larva into segments.

First, the larvae images were converted from RGB to Lab colour space in order to use the L (lightness) channel. Afterwards, for each skeleton point, the average pixel intensity value was determined from the L-channel based on the closest larvae-forming points and a 255-L pixel intensity chart was generated along the determined skeleton. Peaks representing the boundary points of the segments were looked for in the prepared chart. Examples of boundary points representing peaks in the chart are shown in image d in Fig. 5. Based on the boundary points, segments were extracted and classified into head/normal/abdomen end.

To characterise the segments, 25 features were proposed: 12 intensity-related features (mean, skewness, kurtosis, entropy for each histogram R, G, B), 6 texture features (Haralick features<sup>9</sup>: contrast, dissimilarity, homogeneity, energy, correlation, ASM) and 7 shape features (Hu moments<sup>28</sup>). The synthetic minority over-sampling technique (SMOTE)<sup>29</sup> was applied before classification due to unbalanced training data. Selection of the classification model was done by k-fold cross-validation using a training dataset. Finally, the best model was evaluated on an independent test dataset.

A summary of the number of samples in the training and test datasets for the segment classification problem is shown in Table 1. The models examined were logistic regression (LogReg), linear discriminant analysis (LDA)<sup>30</sup>, k nearest neighbours (KNN)<sup>31</sup>, and support vector machines (SVM)<sup>10</sup>. For the KNN and SVM models, hyperparameter optimisation was also performed by checking the number of neighbours for the KNN, and the type of kernel for the SVM. An example of segment classification is shown in image e in Fig. 5. For feature calculation and segment classification, the following Python libraries were used: Scipy<sup>32</sup> (intensity-related features), scikit-image<sup>33</sup> (Haralick features), OpenCV<sup>34</sup> (Hu moments) and scikit-learn<sup>35</sup> (ML models).

**Length distribution estimation.** The overlapping of larvae in the box results in the fact that only parts of some larvae can be seen in the image. If a larva is occluded, it should not be used to estimate the larval length distribution. To obtain a reliable histogram of larvae length in the box, only whole larvae from their head to abdomen end should be extracted from the image. For this purpose, the results of the segment classification described in



"Larvae Phenotyping" section were used. A larva was accepted as proper if the last segments along its skeleton represented the head and abdomen end, respectively. Classification of the last segments as "normal" indicated overlapping.

**Evaluation.** Evaluation was performed for the different tasks included in the monitoring system, namely: object detection (live larvae, pupae, beetles, dead larvae, pests), semantic segmentation (uneaten feed, chitin, frass), the estimation of larval length distribution, and the estimation of larval mass.

*Object detection model evaluation.* For object detection, the predictions and ground truths are in the form of rectangles called bounding boxes, which are described by 4 corner coordinates. Each prediction is also described by a confidence score value, which indicates the percentage of confidence in the prediction. Let us assume that for a given inference, a set of predictions  $P = \{P_1, P_2, \dots, P_n\}$  with confidence score values  $C = \{C_1, C_2, \dots, C_n\}$  were obtained and that the corresponding set of ground truths is  $G = \{G_1, G_2, \dots, G_m\}$ . Moreover, each element from set P and G has the same label depending on the class under consideration. For each bounding box  $G_i$ , let us assign one bounding box  $P_j$  for which: (1) the intersection over union  $IoU(P_j, G_i) \geq 0.5$ , and (2)  $P_j$  has the highest  $C_j$  among the predictions, which satisfies the 1st condition. The number of assignments between G and P is the number of True Positive (TP) predictions. Let us call the number of unassigned predictions from set P as False Positive (FP), and the number of unassigned ground truths from set G as False Negative (FN). From the determined TP, FP, and FN values, the precision and recall metrics can be calculated. Precision PPV (positive predictive value) defines the probability that a given prediction is correct, while recall TPR (true positive rate) defines the probability that a given ground truth object is detected. The formulas for precision and recall are as follows:

$$precision = PPV = \frac{TP}{TP + FP} \quad (4)$$

$$recall = TPR = \frac{TP}{TP + FN} \quad (5)$$

In order to characterise a model by a single value, a metric F1 is often used, which is the harmonic mean of precision and recall. The F1 metric can be calculated using the formula:

$$F1 = \frac{2 * PPV * TPR}{PPV + TPR} \quad (6)$$

Since most of the predictions with a low confidence score are the source of FP errors, some of them that have a value below a certain threshold value  $C_{thresh}$  are removed. On the other hand, too high a value of  $C_{thresh}$  results in more FN errors. The optimal value of  $C_{thresh}$  in such a case may be the value that maximises F1. The value of  $F1_{opt}$ , based on the optimal operating point (threshold value  $C_{opt}$ ), and the related metrics  $PPV_{opt}$ ,  $TPR_{opt}$  were used to evaluate object detection by the proposed models ( $F1_{obj}$ ,  $PPV_{obj}$ ,  $TPR_{obj}$ ). To compare object detection models, it is also good practice to use metrics independent of  $C_{thresh}$ . The most commonly used metric that meets the threshold independence condition is average precision AP. AP is defined as the area under curve (AUC) precision x recall, as represented by the following formula<sup>36</sup>:

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interp}(r_{i+1}) \quad (7)$$

where  $p_{interp}(r_{i+1}) = \max p(\tilde{r})$ , and  $\tilde{r} : \tilde{r} \geq r_{i+1}$ , and  $n$  - the number of predictions.

The precision-recall chart is formed by the precision and recall values at different values of  $C_{thresh}$ . Before calculating the AUC, interpolation of the precision values for the chart points is performed, due to the lack of monotonicity (zigzag shape), with the raw precision x recall chart. For each recall value, the interpolated precision value must be greater than or equal to the precision value for the points with the greater recall value (all points to the right of the considered point), as described in the condition under formula 7. Due to the fact that a bounding box overlap threshold of 50% ( $IoU = 50\%$ ) was assumed, the average precision for detecting objects will be designated as  $AP_{50}$ .

*Semantic segmentation model evaluation.* Semantic segmentation is based on determining a label value for each pixel. Similar to object detection, the metrics  $F1_{opt}$ ,  $PPV_{opt}$ ,  $TPR_{opt}$  for the optimal operating point can be used. The only difference is that individual pixels ( $F1_{pix}$ ,  $PPV_{pix}$ ,  $TPR_{pix}$ ) are considered instead of objects. These three metrics were used in the study to evaluate the semantic segmentation models.

*Larval segment classification model evaluation.* The classification of larval segments was based on the prediction of the head/normal/abdomen end class for each detected segment. Similar to object detection and semantic segmentation, the metrics  $F1_{seg}$ ,  $PPV_{seg}$ ,  $TPR_{seg}$  were used, with each incorrect or correct prediction being associated with one segment.

*Larval length distribution estimation method evaluation.* To validate the methods for estimating larval length distribution, independent test datasets were developed for three populations. Approximately 100 live larvae were

	Type	AP <sub>50</sub>	F1 <sub>obj</sub>	PPV <sub>obj</sub>	TPR <sub>obj</sub>
Live larvae (Tenebrio molitor)	Growth-stage	0.915	0.905	0.927	0.884
Pupae (Tenebrio molitor)		0.900	0.893	0.929	0.859
Beetles (Tenebrio molitor)		0.934	0.930	0.984	0.882
Dead larvae (Tenebrio molitor)	Anomaly	0.814	0.858	0.898	0.821
Pests (Alphitobius diaperinus)		0.777	0.835	0.889	0.787

**Table 2.** Results of the detection growth stages of *Tenebrio molitor* (live larvae, pupae, beetles) and anomalies (dead larvae, pests) for the Mask R-CNN with the ResNet-101 backbone.

selected from each population, and their images were registered. The length of the larvae was determined based on the collected images. A normalised histogram  $h_{true}$  was then determined using the true larval lengths. The estimated normalised histogram  $h_{est}$  was determined from the calculated larval lengths according to the method described in "Larvae Phenotyping" section and by using the filtering of occluded larvae, which was described in "Larvae Phenotyping" section. The formula for the intersection of the histograms was used to determine the similarity of the histograms:

$$D(h_{true}, h_{est}) = \sum_{i=1}^n \min(h_i^{true}, h_i^{est}) \quad (8)$$

where  $D(h_{true}, h_{est})$  - the intersection between two normalised histograms  $h_{true}$  and  $h_{est}$  ( $\sum h_i^{true} = 1$  and  $\sum h_i^{est} = 1$ ), and  $n$  - the number of bins.

Additionally, the means ( $\bar{x}_{true}, \bar{x}_{est}$ ) and standard deviations ( $\sigma_{true}, \sigma_{est}$ ) of the obtained distributions were compared.

**Mass estimation method evaluation.** Evaluation of the method for mass estimation consisted of comparing the true and estimated mass of the larvae in the box. For this purpose, the following experiment was conducted for three different populations. 10 grams of live larvae were added to the box. The total mass of the larvae in the box then ranged from 0 to 100 grams. After each procedure of putting larvae into the box, an image was registered. Each such image was then input to the developed model for instance segmentation to determine the masks for the live larvae. The mass of all the larvae in the box was estimated based on the procedure described in "Volume and mass calculation" section.

The squared Pearson correlation coefficient  $R^2$  between the true mass  $m_{true}$  values and the estimated mass  $m_{est}$  values was used as a quantitative indicator of the estimation quality:

$$R^2 = r_{m_{true}, m_{est}}^2 = \frac{cov^2(m_{true}, m_{est})}{\sigma_{m_{true}}^2 \sigma_{m_{est}}^2} \quad (9)$$

where  $\sigma_{m_{true}}, \sigma_{m_{est}}$  - standard deviations of  $m_{true}$  and  $m_{est}$ , and  $cov(m_{true}, m_{est})$  - the covariance of  $m_{true}$  and  $m_{est}$ .

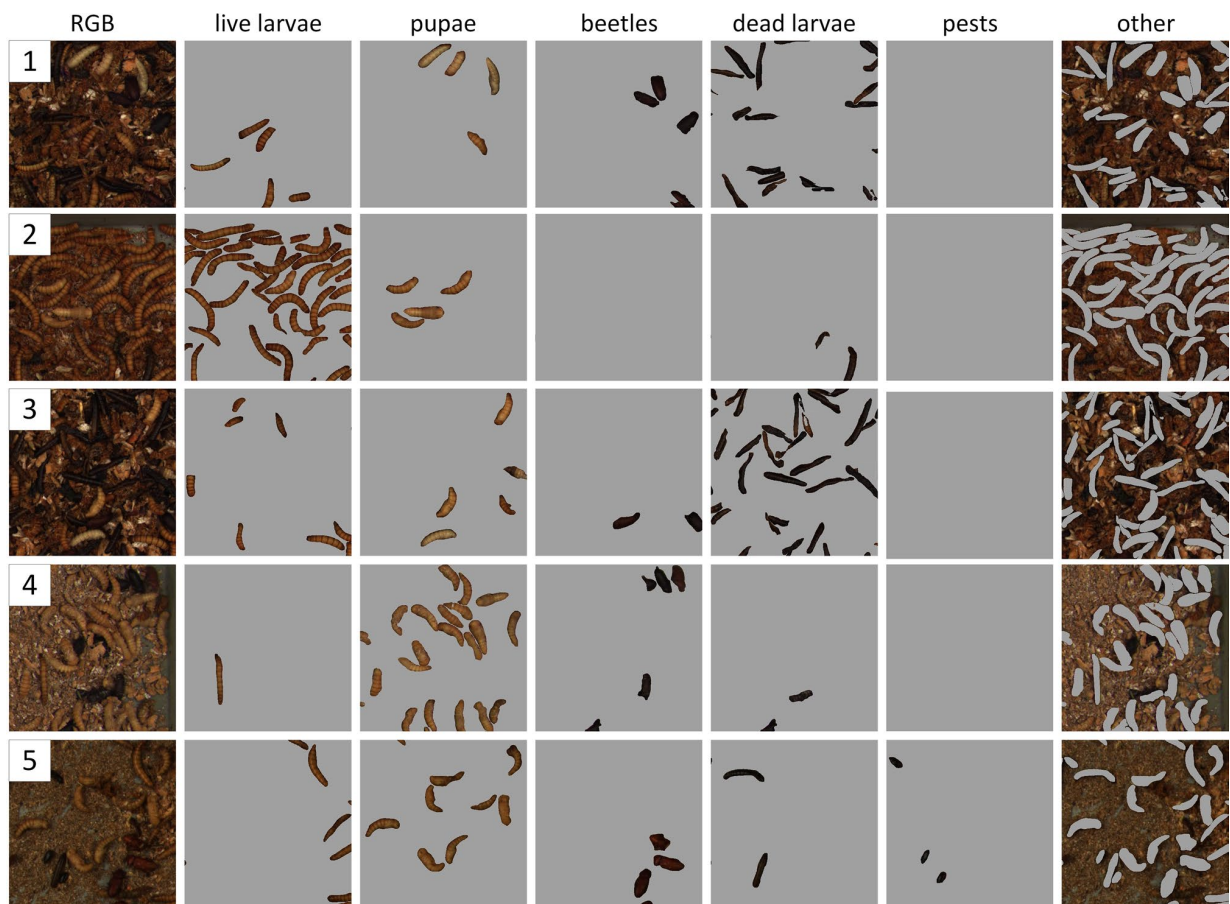
Due to significant overlapping of larvae when there are high numbers of larvae in the box, the  $R^2$  ratio was determined for the initial values of the true larval mass in the box (from 0 to 40 grams). The slope of the line  $a_{0-40}$  in the considered interval was used as an additional parameter of the quality of the mass estimation.

## Results and discussion

This section contains the evaluation results of the different machine learning models, and also the methods used in the proposed modules: ISM, SSM and LPM. The evaluation was performed on test datasets that are independent of the training datasets. Samples for the test datasets were labelled on real images of the boxes with *Tenebrio molitor*. A summary of the number of samples in the training and test datasets is presented in Table 1. An explanation of the metrics used can be found in "Evaluation" section.

The detection results of the *Tenebrio molitor* growth stages (live larvae, pupae, beetles) and anomalies (dead larvae, pests) for the Mask R-CNN model with the ResNet-101 backbone are presented in Table 2 and in Fig. 6.

The proposed instance segmentation model detected growth stages ( $F1_{obj} > 0.89$ ) and anomalies ( $F1_{obj} > 0.83$ ) very efficiently. The density of objects, and their overlapping with each other, did not destructively affect the model's performance—both whole objects and their fragments were detected well (samples 2–4 in Fig. 6 for a high density of live larvae, dead larvae, and pupae, respectively). The robustness of the model to dense scenes is due to the proposed 2D synthetic data generation method described in "Pool-based 2D synthetic image generation" section, where dense scenes with different types of objects that overlap with accurate pixel annotation were simulated. Based on the values of the metrics in Table 2, it can be seen that  $PPV_{obj} > TPR_{obj}$  for all the considered classes. This results in a fewer number of committed FP errors, which comes at the expense of more undetected objects. This choice of operating point is appropriate for anomaly detection, as it reduces the number of possible interventions by the farmer. However, anomalies will mostly be detected anyway due to the presence of more than one object from the anomaly class in the box (samples 3 and 5 in Fig. 6 for dead larvae and pests). Some problems, due to the high cost of non-detection, e.g., detecting the first beetle in the context of breeding interruption, require the operating point to be moved to higher recall values. The few errors made



**Figure 6.** Results of the instance segmentation for live larvae, pupae, beetles, dead larvae, and pests for the sample data.

Model	$AP_{50}$					Inference time/tile
	Live larvae	Pupae	Beetles	Dead larvae	Pests	
Mask R-CNN (ResNet-101)	0.915	0.900	0.934	0.814	0.777	120 ms
YOLOv5x	0.894	0.893	0.921	0.700	0.738	40 ms

**Table 3.** Comparison of object detection  $AP_{50}$  and inference time for the Mask R-CNN and YOLO models.

during inference were mainly: (1) misclassification of object fragments between the live larvae and pupae classes (samples 2–3 in Fig. 6), (2) misclassification of object fragments between the dead larvae and beetles classes (sample 4 in Fig. 6), and (3) undetected object fragments in dense scenes (samples 2–4 in Fig. 6 for the live larvae, dead larvae, and pupae classes, respectively). However, these errors do not affect the usefulness of the proposed ISM.

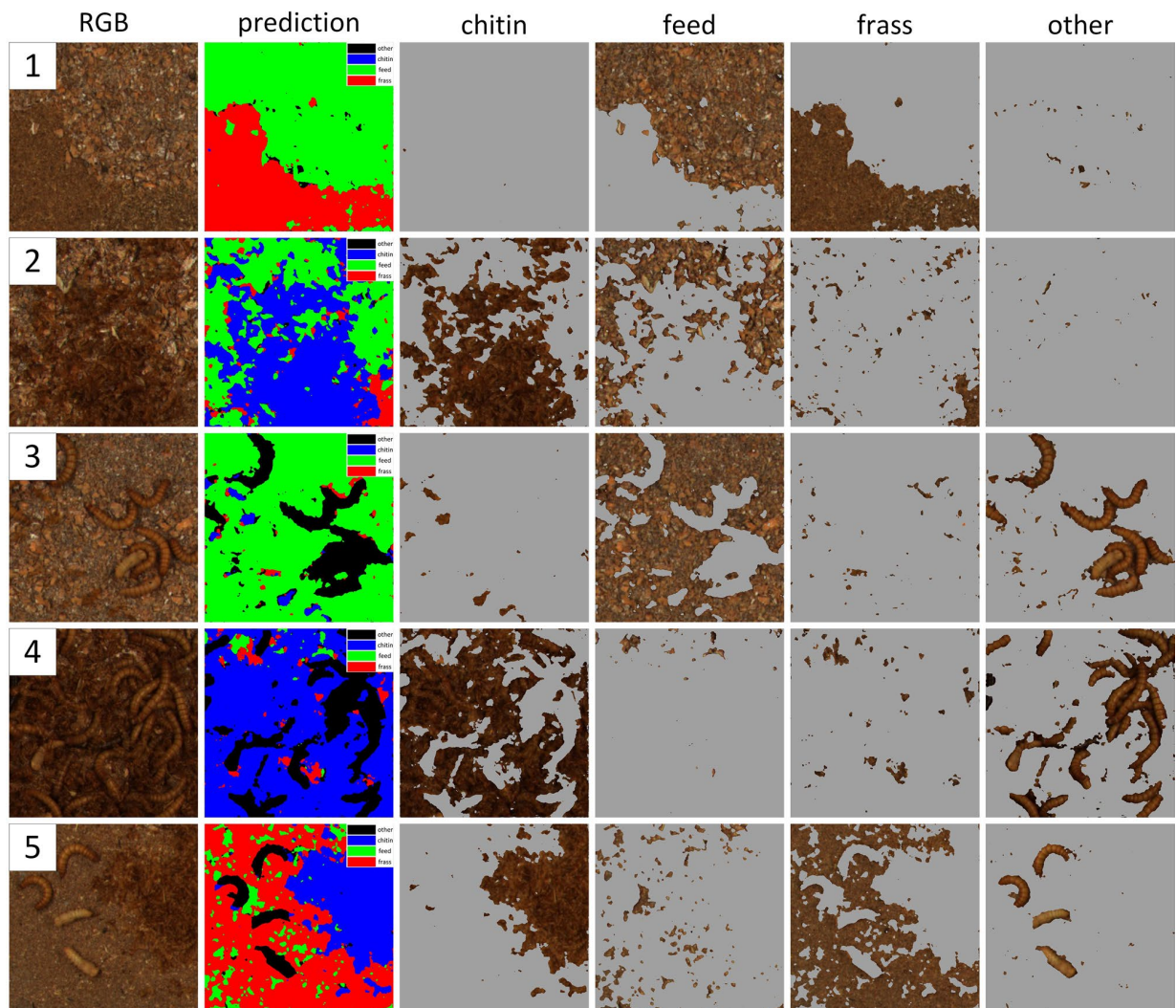
Mask R-CNN, as a representative of an instance segmentation model for detecting different *Tenebrio molitor* growth stages and anomalies, was chosen for this study due to: (1) the ability to further phenotype the detected objects based on their binary masks, (2) the need to extract the instances pixel-wise in order to add them to the object pool (described in "Pool-based 2D synthetic image generation" section), and (3) the simplicity to analyse the prediction and to draw conclusions to further improve the models. Although Mask R-CNN works well in the model development stage, the rationale for its use in the final model should be considered by taking into account the required functionality of the system. If only counting objects from defined classes is required, an object detection model such as YOLO<sup>37</sup> (with significantly less inference time than Mask R-CNN) may be a better solution. As part of the study, the Mask R-CNN with the backbone ResNet-101 model and the YOLOv5x<sup>38</sup> model were also compared in terms of  $AP_{50}$  and inference time for the tile, as shown in Table 3. The GeForce RTX 2060 SUPER 8GB (GPU) and AMD Ryzen 7 1700 3GHz (CPU) were used to measure the inference time for the Mask R-CNN and YOLO models.

The  $AP_{50}$  values in Table 3 for all the considered classes (except the dead larvae), when comparing the Mask R-CNN (ResNet-101) and YOLOv5x models, decreased slightly. Moreover, the inference time decreased about three times. Taking into account that the RGB image of the whole box (example shown in Fig. 1) was split into



	$F1_{pix}$	$PPV_{pix}$	$TPR_{pix}$
Feed	0.971	0.969	0.973
Chitin	0.947	0.918	0.977
Frass	0.953	0.963	0.943

**Table 4.** Results of the semantic segmentation for the feed, chitin and frass for the U-Net with the EfficientNet-B0 backbone.



**Figure 7.** Results of the semantic segmentation of the feed, chitin and frass for the sample data.

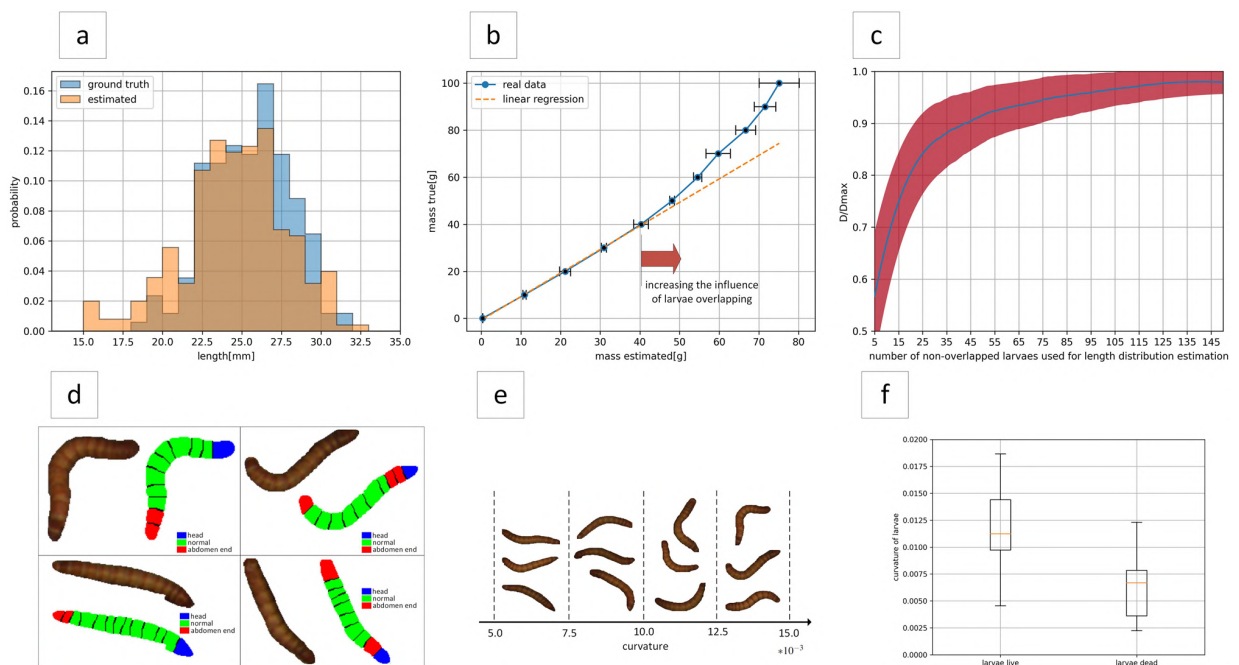
192 standard size tiles before inference, and the fact that the total inference time is notable, use of the YOLOv5x model in the final monitoring system can be seen to be appropriate.

The semantic segmentation results for the U-Net model with the EfficientNet-B0 backbone are presented in Table 4 and in Fig. 7.

In the case of the semantic segmentation, the achieved values of  $F1_{pix} > 0.94$  for the feed, chitin and frass classes demonstrate the ability of the model to efficiently segment the regions that represent the defined classes. The SSM was able to cope with both the segmentation of the larger regions of a class, as well as smaller regions, e.g., one feed flake, one chitin moult, as can be observed on samples 1–5 in Fig. 7. The most common inference errors for the SSM were: (1) mistakes between the feed and frass classes (samples 1 and 5 in Fig. 7) for areas without cereal grains, (2) chitin segmentation at the end segments (head, abdomen end) of the live larvae (sample 5 in Fig. 7), and (3) mistakes in the small areas between objects e.g. larvae (samples 3 and 4 in Fig. 7). For the SSM, it is important to note the very fast development process of the model. The proposed labelling method

Segment type	Model	$TPR_{seg}$		$PPV_{seg}$		$F1_{seg}$	
		Raw predictions	Processed predictions	Raw predictions	Processed predictions	Raw predictions	Processed predictions
Normal	LDA	0.895	0.983	0.988	0.989	0.939	0.986
	LogReg	0.930	0.990	0.996	0.996	0.962	0.993
	SVM	0.920	0.986	1.000	1.000	0.958	0.993
Head	LDA	0.833	0.833	0.600	0.769	0.698	0.800
	LogReg	0.750	0.750	0.692	0.794	0.720	0.771
	SVM	0.750	0.750	0.692	0.794	0.720	0.771
Abdomen end	LDA	0.750	0.750	0.551	0.771	0.635	0.760
	LogReg	0.806	0.806	0.558	0.725	0.659	0.763
	SVM	0.833	0.833	0.536	0.714	0.652	0.769
All (average metrics)	LDA	0.826	0.855	0.713	0.843	0.757	0.849
	LogReg	0.829	0.849	0.749	0.838	0.780	0.842
	SVM	0.834	0.856	0.743	0.836	0.777	0.844

**Table 5.** Results of the classification of the segments of the live larvae into head, normal and abdomen end.



**Figure 8.** Charts related to larval phenotyping: (a) comparison of true and estimated larval length distribution for a selected population, (b) estimation of larvae mass based on averaged samples, (c) chart of the normalised intersection of histograms as a function of the number of non-overlapping larvae used to estimate the larval length distribution, (d) segment classification for example larvae using the best proposed model, (e) examples of larvae for specific ranges of curvature values, (f) comparison of curvature for live and dead larvae.

described in "Labelling" section (splitting a larger annotated area into smaller ones and adding them to a pool), together with the generation of synthetic samples described in "Pool-based 2D synthetic image generation" section, enabled a significant reduction in the model's development time. Moreover, the model maintained a comparable diversity of the samples and an increase in label veracity when compared to the classical labelling of the samples for the semantic segmentation.

The classification of larval segments was one of the components of the LPM. Evaluation results on the test dataset for larval segment classification for the top three models (LDA, SVM, LogReg) were shown in Table 5. In Fig. 8, selected errors made by the classifiers can be analysed. Most of the incorrect predictions are related to the classification of segments located before the end segments. By analyzing the characteristics of these mistakes, it is easy to eliminate them by treating duplicate head or abdomen end predictions of neighbouring segments as one prediction. In Table 5, it can be seen that the proposed prediction processing method taking segment location into account significantly increased the examined metrics for all shown models. Considering the averaged metrics

Population id	$\bar{x}_{est}$ [mm]	$\sigma_{est}$ [mm]	$\bar{x}_{true}$ [mm]	$\sigma_{true}$ [mm]	$D(h_{true}, h_{est})$	$R_{0-40}^2$	$a_{0-40}$
1	27.4	3.1	27.8	3.3	0.806	0.998	1.02
2	24.6	3.3	25.5	2.6	0.842	0.999	0.95
3	26.5	2.3	26.2	1.9	0.874	0.999	1.04

**Table 6.** Results of estimating the larval length distribution and mass for the three study populations.

x	% of box area	ISM		SSM		LPM		Total time per box	No. of boxes per day
		Model	Time per box	Model	Time per box	Parameters	Time per box		
1	100 (192 tiles)	Mask R-CNN	30.6 s	U-Net	8.7 s	Proposed all	186.4 s	225.7 s	380
2	25 (12 tiles)	Mask R-CNN	2.0 s	U-Net	0.5 s	Proposed geometric (length, volume, curvature)	52.4 s	54.9 s	1570
3	25 (12 tiles)	Mask R-CNN	2.0 s	U-Net	0.5 s	Basic (area of binary mask)	0.4 s	2.9 s	30,000
4	25 (12 tiles)	YOLOv5x	1.1 s	U-Net	0.5 s	–	–	1.6 s	54,000

**Table 7.** Computational burden analysis for four versions of the proposed system.

after prediction processing, the best model for segment classification was LDA, which was characterized by  $F1_{score} > 0.75$  for each considered class: head/normal/abdomen end. Its usefulness in filtering whole larvae from fragments was confirmed by the high intersection values of the larval length histograms ( $D(h_{true}, h_{est}) > 0.8$ ).

The results obtained for the estimation of the larvae length distribution (Table 6 and Chart a in Fig. 8) prove that the developed method for determining these quantities is efficient. The high histogram similarity values obtained ( $D(h_{true}, h_{est}) > 0.8$ ) exceed the requirements for a monitoring system. Taking into account that the number of visible larvae in a box can reach 1000, it is necessary to consider the validity of phenotyping all visible larvae, which is computationally expensive. To this purpose, a chart was prepared of the normalised intersection of histograms (relative to the maximum intersection value obtained when using all larvae for estimation) as a function of the number of non-overlapping larvae used to estimate the larval length distribution, which is shown in Chart c in Fig. 8. This chart shows that adding more larvae (above 45 individuals) to the estimation no longer contributes significantly to the histogram intersection, while 45 individuals allows a value of about 0.9 of the maximum histogram intersection to be achieved, which is definitely an acceptable compromise. From Chart a in Fig. 8, it can be observed that histogram mismatches occur mainly in the tails of the distributions: (1) the underestimation of the number of longer larvae (right tail) results from the higher probability of such larvae being occluded under breeding conditions, while (2) the overestimation of the number of shorter larvae (left tail) results from the few errors during segment classification and the fact that occluded larvae are taken for estimation. The occurrence of the mentioned problems does not negate the usefulness of the proposed method for estimating the larval length distribution for larval growth monitoring during breeding.

The results for the larvae mass estimation problem in Table 6, and Chart b in Fig. 8, confirm that the proposed method for estimating the volume and mass of larvae is appropriate, as indicated by the values of the metrics for the range from 0 to 40 grams ( $R_{0-40}^2 > 0.99$  and  $|1 - a_{0-40}| < 6\%$ ). However, its applicability under breeding conditions with high larval overlap is limited. This method can mainly be seen to have potential in experiments with relatively small numbers of larvae, e.g. testing new types of feed in laboratory breeding studies, when it allows for the non-invasive determination of larval weight gains. A solution for an effective vision-based determination of larval weight in the box under real breeding conditions may be a hybrid approach. Using the knowledge of the larval length distribution, larval growth stage, and approximate number of individuals per box (at the beginning of breeding it is similar for all boxes), a model can be developed to also estimate the mass of unseen larvae. Verification of this idea is the next direction of our research.

The proposed curvature parameter may be one of the indicators of larvae health, so it was included in the LPM. Images of larvae with different curvature values are shown in Chart e in Fig. 8. The stiffening phenomenon (reduction in the value of the curvature parameter) was observed when the larvae die, as shown in Box chart f in Fig. 8, which compares the curvature distribution for live and dead larvae. The stiffening phenomenon could also be observed in the case of larvae/pupae transformation. Preliminary studies show the potential of the curvature parameter for larval phenotyping, but investigating its usefulness is a topic for further studies based on long-term observations.

The obtained evaluation results of the developed modules give attitudes to believe that the proposed system can be used in a real scenario. The study proposed four versions of the system usage depending on the needs of the user (researcher or breeder), as shown in Table 7. The first (full version) of the system assumes accurate image analysis of the entire box (192 tiles in total, which includes additional tiles for reducing edge effect related to the difficulty of detecting objects on the edges of a tile) using Mask-RCNN for ISM and U-Net for SSM. Phenotyping in the first version includes computation of all proposed features for selected 50 larvae from the box. The second version of the system includes the analysis of 25 percent of the box area (without reducing edge effects) using the same models as in the first version. Phenotyping in the second version includes computation of the proposed geometric features, i.e. length, volume, and curvature, without division into segments, and their classification. The

third version limits the phenotyping of larvae only to the basic parameter of the binary mask area and the rest of the assumptions are the same as in the second version. In the fourth version, larval phenotyping is dispensed with, allowing the model to be changed from Mask R-CNN to YOLOv5x in the ISM module. The presented versions of the system represent a trade-off between the amount of population information obtained and the inference time (the number of boxes that can be analyzed per day) and the rationale of their use depends on the needs of the user. Undoubtedly, options 3 and 4 can be considered for use in monitoring large-scale edible insect breeding (30,000 and 54,000 analyzed boxes per day). The use of options 1 and 2 should be seen in monitoring smaller farms and for laboratory breeding studies. It should also be noted that larval phenotyping is the bottleneck of the whole system and future work should focus on increasing the efficiency of the LPM module.

The application of our system in a breeding environment requires the preparation of appropriate hardware and software architecture. Potential users of the system should pay attention to the following recommendations: (1) computer with GPU (or external server with GPU) enabling fast prediction by proposed ML models, (2) periodic automated boxes inspection (frequency depending on requirements, initially once every day or two days may be assumed), (3) database containing calculated features for each analyzed box, (4) identification of individual boxes with edible insects (e.g. RFID), (5) application (Web, mobile) for the farmer, including reporting of anomalies and with the possibility to view historical data (changes of characteristics over time for a specific box), (6) additional system to control and optimize the breeding process based on current data.

## Conclusions

The developed multipurpose monitoring system for the breeding of *Tenebrio molitor* based on three modules (ISM, SSM and LPM) has great potential for the observation of edible insects in both laboratory breeding studies and real breeding conditions. The proposed method for developing multiclass instance and semantic segmentation models based on synthetic image generation and object pools significantly reduced the time of the iterative improvement of machine learning models, while also increasing the robustness of the models to problems such as dense scenes and the detection of minority class objects. The described method for estimating the length distribution of larvae in a box enables effective supervision of larval growth during breeding, even when most larvae are invisible. The developed larval mass estimation methods can be successfully applied to feeding experiments for the non-invasive assessment of mass gains. Future work will include: (1) the improvement of the synthetic image generation process and the quality of generated images, (2) the improvement in efficiency (inference time reduction) for the larvae phenotyping module, (3) the extension of the larvae phenotyping module to include more features for the growth stages of *Tenebrio molitor* (also for pupae and beetles), (4) the development of a module to determine larval activity using temporal data based on optical flow, (5) the long-term observation of *Tenebrio molitor* in order to characterise the change in states (live larvae to dead larvae, live larvae to pupae) and behavioural patterns, (6) the association of larval features with symptoms of disease or poor condition, and (7) re-identification for individual larvae and beetles.

## Data availability

The samples used to develop the ISM, SSM and LPM modules and the generated sample synthetic data are available from the corresponding author on reasonable request.

Received: 2 March 2022; Accepted: 25 April 2022

Published online: 12 May 2022

## References

1. Joint, F., Organization, W. H. *et al.* *Protein and Amino Acid Requirements in Human Nutrition: Report of a Joint FAO/WHO/UNU Expert Consultation* (World Health Organization, 2007).
2. Thrastardottir, R., Olafsdottir, H. T. & Thorarinsdottir, R. I. Yellow mealworm and black soldier fly larvae for feed and food production in Europe, with emphasis on Iceland. *Foods* **10**, 2744 (2021).
3. Miryam, D., Bar, P. & Oscherov, M. Ciclo de vida de *tenebrio molitor* (coleoptera, tenebrionidae) en condiciones experimentales [life cycle of *tenebrio molitor* (coleoptera, tenebrionidae) under experimental conditions]. *Methods* (2000).
4. Bakula, T. & Gałęcki, R. In *Strategia wykorzystania owadów jako alternatywnych źródeł białka w żywieniu zwierząt oraz możliwości rozwoju jego produkcji na terytorium Rzeczypospolitej Polskiej [Strategy of using insects as alternative sources of protein in animal feed and the possibility of developing its production in the territory of the Republic of Poland]*, 261–315 (2021).
5. Wang, J., Yue, H. & Zhou, Z. An improved traceability system for food quality assurance and evaluation based on fuzzy classification and neural network. *Food Control* **79**, 363–370 (2017).
6. Migliozzi, D. *et al.* Multimodal imaging and high-throughput image-processing for drug screening on living organisms on-chip. *J. Biomed. Opt.* **24**, 021205 (2018).
7. Tao, D., Wang, Z., Li, G. & Xie, L. Sex determination of silkworm pupae using vis-nir hyperspectral imaging combined with chemometrics. *Spectrochim. Acta Part A Mol. Biomol. Spectrosc.* **208**, 7–12 (2019).
8. Araújo, M. C. U. *et al.* The successive projections algorithm for variable selection in spectroscopic multicomponent analysis. *Chemom. Intell. Lab. Syst.* **57**, 65–73 (2001).
9. Haralick, R. M., Shanmugam, K. & Dinstein, I. H. Textural features for image classification. *IEEE Trans. Syst. Man Cybern.* 610–621 (1973).
10. Cortes, C. & Vapnik, V. Support-vector networks. *Mach. Learn.* **20**, 273–297 (1995).
11. Sumriddetchkajorn, S., Kamtongdee, C. & Chanhorm, S. Fault-tolerant optical-penetration-based silkworm gender identification. *Comput. Electr. Agric.* **119**, 201–208 (2015).
12. Sabour, S., Frosst, N. & Hinton, G. E. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829* (2017).
13. Agarwal, M. *et al.* Identification and diagnosis of whole body and fragments of trogoderma granarium and trogoderma variabile using visible near infrared hyperspectral imaging technique coupled with deep learning. *Comput. Electron. Agric.* **173**, 105438 (2020).
14. Cook, D. F., Voss, S. C. & Dadour, I. R. The laying of live larvae by the blowfly *Calliphora vicina* (Diptera: Calliphoridae). *Forensic Sci. Int.* **223**, 44–46 (2012).



15. Kröncke, N. *et al.* Automation of insect mass rearing and processing technologies of mealworms (*tenebrio molitor*). In *African Edible Insects As Alternative Source of Food, Oil, Protein and Bioactive Components*, 123–139 (Springer, 2020).
16. Wang, J. *et al.* Semi-supervised active learning for instance segmentation via scoring predictions. *arXiv preprint arXiv:2012.04829* (2020).
17. Toda, Y. *et al.* Training instance segmentation neural network with synthetic datasets for crop seed phenotyping. *Commun. Biol.* **3**, 1–12 (2020).
18. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, 2961–2969 (2017).
19. Ren, S., He, K., Girshick, R. & Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **28**, 91–99 (2015).
20. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).
21. Machefer, M., Lemarchand, F., Bonnefond, V., Hitchins, A. & Sidiropoulos, P. Mask r-cnn refitting strategy for plant counting and sizing in uav imagery. *Remote Sens.* **12**, 3015 (2020).
22. Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y. & Girshick, R. Detectron2. <https://github.com/facebookresearch/detectron2> (2019).
23. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234–241 (Springer, 2015).
24. Tan, M. & Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 6105–6114 (PMLR, 2019).
25. Deng, J. *et al.* Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255 (IEEE, 2009).
26. Yakubovskiy, P. Segmentation models. [https://github.com/qubvel/segmentation\\_models](https://github.com/qubvel/segmentation_models) (2019).
27. Zhang, T. Y. & Suen, C. Y. A fast parallel algorithm for thinning digital patterns. *Commun. ACM* **27**, 236–239 (1984).
28. Hu, M.-K. Visual pattern recognition by moment invariants. *IRE Trans. Inf. Theory* **8**, 179–187 (1962).
29. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002).
30. Fisher, R. A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **7**, 179–188 (1936).
31. Fix, E. & Hodges, J. L. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique* **57**, 238–247 (1989).
32. fundamental algorithms for scientific computing in python. Virtanen, P. *et al.* Scipy 1.0. *Nat. Methods* **17**, 261–272 (2020).
33. Van der Walt, S. *et al.* scikit-image: image processing in python. *PeerJ* **2**, e453 (2014).
34. Bradski, G. The opencv library. *Dr. Dobbs's J. Softw. Tools Prof. Programm.* **25**, 120–123 (2000).
35. Pedregosa, F. *et al.* Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
36. Padilla, R., Passos, W. L., Dias, T. L., Netto, S. L. & da Silva, E. A. A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics* **10**, 279 (2021).
37. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788 (2016).
38. Jocher, G., Nishimura, K., Mineeva, T. & Vilariño, R. Yolov5. <https://github.com/ultralytics/yolov5> (2020).

## Acknowledgements

We wish to thank Mariusz Mrzygłód for developing applications for the designed data acquisition workstation. We wish to thank Paweł Górzynski and Dawid Biedrzycki from Tenebria (Lubawa, Poland) for providing a data source of boxes with *Tenebrio molitor*. The work presented in this publication was carried out within the project “Automatic mealworm breeding system with the development of feeding technology” under Sub-measure 1.1.1 of the Smart Growth Operational Program 2014-2020 co-financed from the European Regional Development Fund on the basis of a co-financing agreement concluded with the National Center for Research and Development (NCBiR, Poland); grant POIR.01.01.01-00-0903/20.

## Author contributions

Conceptualisation, P.M., J.R.; methodology, P.M., P.L.; software, P.M.; validation, P.M.; formal analysis, P.M.; investigation, P.M.; resources, P.Z.; data curation, P.M., P.L.; writing—original draft preparation, P.M., P.Z.; writing—review and editing, P.M., P.Z., P.L., R.B., J.R.; visualisation, P.M., P.L.; supervision, J.R., R.B.; project administration, J.R.; funding acquisition, J.R.

## Competing Interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to P.M.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022



## **4.2 Prediction of the remaining time of the foraging activity of honey bees using spatio-temporal correction and periodic model re-fitting**

**Authors:** Paweł Majewski, Piotr Lampa, Robert Burduk, and Jacek Reiner

**Publication status:** published

**Type of publication:** journal paper

**Journal/Conference:** Computers and Electronics in Agriculture (IF=8.3)

**MEiN points:** 100

**Lead Author:** Yes

**Corresponding Author:** Yes

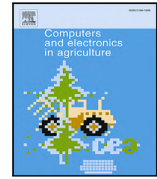
**Percentage contribution:** 60%

**CRedit:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft preparation



Contents lists available at ScienceDirect

## Computers and Electronics in Agriculture

journal homepage: [www.elsevier.com/locate/compag](http://www.elsevier.com/locate/compag)

Original papers

## Prediction of the remaining time of the foraging activity of honey bees using spatio-temporal correction and periodic model re-fitting

Paweł Majewski<sup>a,\*</sup>, Piotr Lampa<sup>b</sup>, Robert Burduk<sup>a</sup>, Jacek Reiner<sup>b</sup><sup>a</sup> Faculty of Information and Communication Technology, Wrocław University of Science and Technology, Poland<sup>b</sup> Faculty of Mechanical Engineering, Wrocław University of Science and Technology, Poland

## ARTICLE INFO

## Keywords:

Honey bee  
Foraging activity  
Machine learning  
Computer vision  
Concept drift  
IoT

## ABSTRACT

The problem of bee poisoning causes significant losses to the beekeeping sector every year. One cause of bee poisoning is spraying before the end of the foraging activity of bees. Information about the estimated end of this foraging activity can significantly help a farmer plan his spraying. The aim of our research was to develop a method based on machine learning models to predict the remaining time of the foraging activity, taking into account bee activity, weather conditions, and the amount of time to sunset. Data were collected using an IoT system from 3 hives in the 2021 and 2022 beekeeping seasons. The proposed method addresses the challenge of the changing nature of data during the beekeeping season by using periodic model re-fitting with automatically generated semi-true target values. The veracity of semi-true target values was also improved by a spatio-temporal correction mechanism based on the position and orientation of the bees, which made it possible to distinguish foraging from other patterns of bee behavior (dead bees, hive ventilation by bees). The results of the RMSE prediction error of 23.1 min (season 2021) and 26.5 min (season 2022) prove the high potential of the proposed method to predict the remaining time of the foraging activity of bees, as well as the lack of need for expert annotation of data during the season. The used approach, based on density occurrence maps in spatio-temporal correction, can also be used in the future to detect and study bee behavior patterns.

## 1. Introduction

The need to increase food production for an ever-growing human population means that the intensification of agriculture is unavoidable. Plant protection products are undoubtedly necessary for the effective control of pests and weeds, and also for the prevention of diseases, but their improper use can lead to environmental degradation.

In the age of agricultural intensification, bee poisoning is not uncommon. According to a report by the Apiculture Division in Puławy (Poland), more than 25,000 bee colony poisonings were reported in Poland in 2020 (Semkiw, 2020), the main cause of which was the spraying of rapeseed (especially spraying against the rapeseed pollen beetle, *Brassicoglyphus aeneus* at the wrong time, including during the flowering of crops or associated weeds, and also when the bees had not finished foraging. Losses caused by bee poisoning are mainly connected with: (1) the lack, or reduction, in the amount of obtained bee products (honey, bee pollen, wax, propolis, royal jelly); (2) the collapse, weakening, or inhibition of the development of bee colonies; and (3) the failure of bees to pollinate crops (Skubida, 2007). The reasons for the occurrence of bee poisoning include: (I) poor communication between the beekeeper and the farmer; (II) the lack of a uniform system

that supports the farmer's decision on the possibility of spraying; and (III) unclear legal regulations. A comprehensive advisory system with a platform for information exchange between the farmer and beekeeper could help in terms of making win-win decisions. The operation of the system would be based on the registration of spraying that is planned (carried out) by the farmer, and would take into account the location of the farmer's field, the type of crop, and the type of used pesticide. The system would also have information about: (1) legal regulations; (2) expert knowledge and good agricultural practices; (3) the location of apiaries in the region; and (4) the current state of apiaries in terms of ongoing bee foraging activity. Taking into account the input data, the system would suggest the optimal time for the farmer to carry out the spraying. An important part of such a system would be a model for the prediction of the time remaining to the end of bee foraging activity, which would in turn allow the farmer to schedule spraying in advance and maximize the time between spraying and the start of bee activity for the following day. The prediction model would use processed raw data from the IoT system (e.g. images, weather data). With up-to-date information from the apiary, the problem of poisoning

\* Corresponding author.

E-mail address: [pawel.majewski@pwr.edu.pl](mailto:pawel.majewski@pwr.edu.pl) (P. Majewski).<https://doi.org/10.1016/j.compag.2022.107596>

Received 14 June 2022; Received in revised form 26 September 2022; Accepted 26 December 2022

Available online 3 January 2023

0168-1699/© 2022 Elsevier B.V. All rights reserved.

caused by spraying too early, when the bees have not yet finished their foraging activity, would also be eliminated.

Researchers have often addressed the use of computer vision and machine learning methods to monitor bees. The first such studies concerned the detection of bees at the entrance to the hive. For bee detection, both classical computer vision methods (Campbell et al., 2008), and newer models for object detection based on deep convolutional networks (Ryu et al., 2021; Dembski and Szymański, 2020) were used. In the literature, studies dedicated to issues such as detecting bee pollen loads (Rodríguez et al., 2018; Stojnić et al., 2018), the cell classification of bee frames (Alves et al., 2020), the detection of Varroa destructor parasites on bees (Bjerger et al., 2019), the tracking of bees (Bozek et al., 2021; Ngo et al., 2019; Bozek et al., 2018), and the re-identification of bees (Chan et al., 2022) can also be found. It is also worth noting that there are papers on the development of IoT systems for apiaries, which enable their monitoring in real-time (Ngo et al., 2021a; Marsteller et al., 2019; Tashakkori et al., 2021).

A much smaller number of papers concern the analysis of long-term bee activity and the prediction of bee behavior. Gomes et al. (2020) predicted bee foraging activity using recurrent neural networks based on a time series of activity level, temperature, solar radiation, and barometric pressure within a time window of a specified length. The researchers used RFID tagging of bees to record their activity. The activity level was calculated for each hour, and the optimal time window size was 24 h. A significant limitation of the method proposed in this paper is the recording of bee activity through RFID tagging. This cannot be applied to noninvasive IoT systems, which should be the basis of apiary monitoring. Ngo et al. (2021b) predicted daily bee losses using temporal convolutional networks (Lea et al., 2016) based on bee activity (represented by the number of bees entering and leaving the hive), temperature, humidity, and wind and rainfall-related features. Clarke and Robert (2018) modeled the foraging activity of bees (the bee egress rate) based on temperature, solar radiation, atmospheric pressure, humidity, rainfall, wind direction, and speed using the ordinary-least-squares model. The authors reported that 78% of the observed variation in bee activity was explained by variations in temperature and solar radiation. Andrijević et al. (2022) modeled the hourly activity of bees entering and exiting the hive using multidomain characteristics collected inside and outside the hive. The researchers used ARIMA (Box et al., 2015), Prophet (Taylor and Letham, 2018), and LSTM (Hochreiter and Schmidhuber, 1997) models in order to develop prediction models. A bee counting sensor array mounted at the entrance to the hive was used to record activity. Undoubtedly, the approach presented in this paper, with the monitoring of multiple environmental factors, can be seen to be reasonable from a research perspective. However, when designing systems to support the beekeeper's decision, one must consider the trade-off between the cost and invasiveness of sensors and the quality of information obtained by the beekeeper.

Although the work described above has demonstrated the possibility and potential of using computer vision and machine learning to address issues of long-term bee monitoring and prediction, the researchers did not explicitly explore the problem of maintaining high model performance during the entire beekeeping season. Considering the dynamic nature of bee colony development, as well as changing weather conditions, it is expected that the character of the input data for the models will change significantly, with the development of adaptation mechanisms being crucial in the context of the developed solutions. The described phenomenon of changing the distribution of data over time is called concept drift, and methods related to responding to concept drift for streaming data are a current research topic, also in the field of insect observations (Rustia et al., 2021; de Souza et al., 2013). In the literature, studies related to the direct prediction of the end time of bee foraging activity, which is important information from the point of view of a farmer who wants to spray, were not found. It should also be noted that a significant limitation of some of the proposed solutions are IoT systems that significantly interfere with the design of a particular

hive, or with the daily functioning of the bees. Minimizing the impact of an IoT system on these two aspects should be a key consideration when developing monitoring systems for an apiary.

The aim of our work was to develop an efficient and robust method for predicting the time remaining to the end of the daily foraging activity of bees. The method takes into account the varying nature of the input data, which is based on data acquired from an IoT system. The main achievements of our work are: (1) the development of a model for predicting the remaining time of the foraging activity of bees, which takes into account bee activity, weather conditions, and the amount of time to sunset; (2) the development of a mechanism to maintain the quality of the models for long-term observation (the duration of the beekeeping season), using periodic model re-fitting with automatically generated semi-true target values; (3) the proposed method for spatio-temporal correction taking into account the location and orientation of bees that reduces the error of semi-true target values determination; (4) a modular, non-invasive and versatile IoT system enabling real-time data collection and analysis; and (5) multi-faceted validation and parameter fine-tuning of the proposed methods based on data from the 2021 and 2022 beekeeping seasons.

## 2. Materials and methods

This section addresses the following topics successively: (1) the definition of the problem and the scheme of the proposed solution; (2) the development of the data acquisition station; (3) the characteristics of the collected data; (4) methods of detecting bees and determining their orientation; (5) regression models for predicting the remaining time of the foraging activity of bees; (6) the initial model fitting and periodic model re-fitting strategy; (7) the spatio-temporal correction of the registered number of bees; (8) fine-tuning of the parameters for the proposed methods; and (9) the types of metrics used in the evaluation of the proposed methods.

### 2.1. Definition of the problem

The considered problem is the prediction of the time remaining to the end of the daily foraging activity of bees, and takes into account the following features:

1. the time remaining until sunset  $\Delta t_{sunset}$  (the sunset time is known for each calendar day),
2. the daily bee activity  $[a_1, a_2, \dots, a_n]$  in a time window of length  $t_{chunk}$ ,
3. weather characteristics (temperature  $[T_1, T_2, \dots, T_n]$ , humidity  $[\phi_1, \phi_2, \dots, \phi_n]$ , and barometric pressure  $[P_1, P_2, \dots, P_n]$ ) in a time window of length  $t_{chunk}$ .

The scheme of the proposed solution is shown in Fig. 1. The next steps of developing the method are described in the following sections.

### 2.2. Data acquisition station

The designed acquisition station (Fig. 2) enabled images from the hive entrance to be captured, and also the weather conditions (temperature, humidity, barometric pressure) at a given sampling period to be recorded. The image acquisition modules were mounted on 3 hives, and the weather sensor was placed near them in a radiation shield at a height of approximately 1.5 m. Images with a resolution of  $1920 \times 1080$  pixels were collected using Raspberry Pi Cameras V2, which are controlled by SBCs (Single Board Computers) and Raspberry Pi 4B. The SBCs were connected to a local WiFi network, which was in turn connected to the Internet. For the described study, data were collected every 1 min from 3 hive-mounted stations from sunrise to sunset with an offset of 1 h. The collected data were uploaded to a cloud database once a day. The hardware part of the IoT system was non-invasive (both to the bees' functioning and to the hives' construction) and modular (ease of replicating the station for more hives).

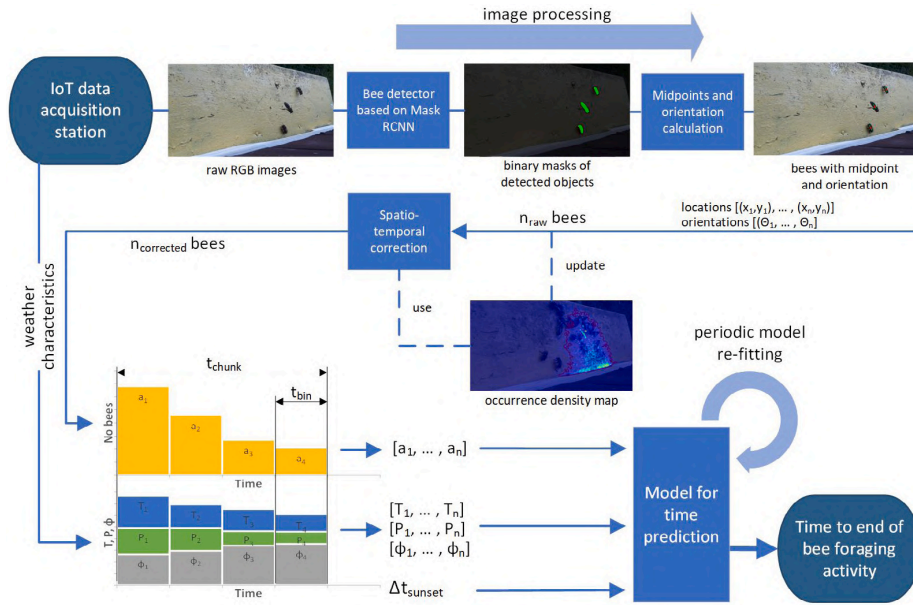


Fig. 1. Scheme for the proposed solution.

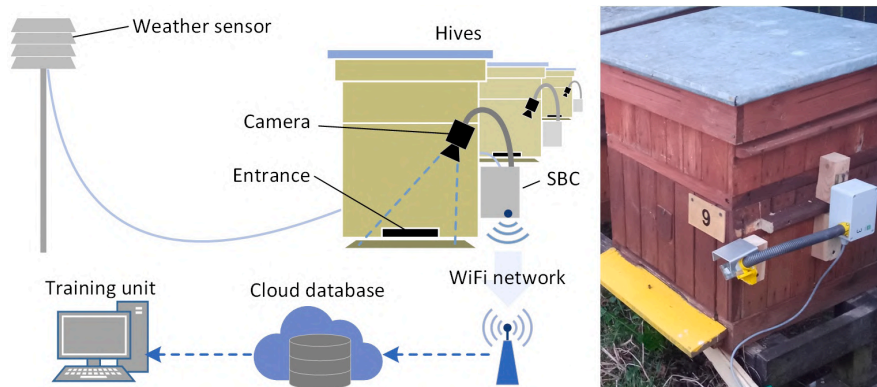


Fig. 2. Data acquisition station.

2.3. Data

The data used in the study were collected in the 2021 and 2022 beekeeping seasons. The end of the foraging activity of the bees was analyzed from 26 April to 29 June in the 2021 season, and from 12 April to 5 June in the 2022 season. For each day and for each hive monitored, an expert determined the end of the bee foraging activity using a sequence of images collected on that day. Samples collected during rainy days were not used for annotation due to the fact that there is no bee foraging and the farmer is not able to spray. A summary of the end times of the bee foraging activity for the 2021 and 2022 seasons is presented in Fig. 3.

The raw data collected from each station was divided into chunks, which were then used by the proposed machine learning models for training and prediction. Each chunk was characterized by a specific length  $t_{chunk}$ , which determines what historical data should be included in the chunk. To eliminate noise and to reduce dimensionality, instead of using raw feature values (number of bees, weather indicators), averaged values in bins of a specific length  $t_{bin}$  were used, where  $t_{bin} < t_{chunk}$ . In this study, chunks were used in which the youngest

observation was recorded no earlier than 6 h before sunset. An explanation of the pre-processing and the parameters  $t_{chunk}$ ,  $t_{bin}$  is also provided in Fig. 1.

2.4. Bee detection and orientation determination

Bee detection was performed using the Mask R-CNN (He et al., 2017) model, which was trained on samples of real images. Images differing in acquisition time (e.g. early morning, evening), bee density, the presence of overexposure, and the bee growth stage were selected for the training set. Labeling involved manually drawing polygons for subsequent instances using labelme software (Wada, 2018). In total, 143 images containing 1047 labeled bee instances were used for the training. The bee detection model was validated on a test set that contained 37 images (211 labeled bee instances). Samples from different days were selected for the training and test sets in order to ensure independence between these sets. ResNet50 (He et al., 2016) was used as the backbone for the Mask R-CNN. The obtained binary masks after Mask R-CNN inference allowed for the determination of the midpoint and orientation for the detected bees. The orientation was determined



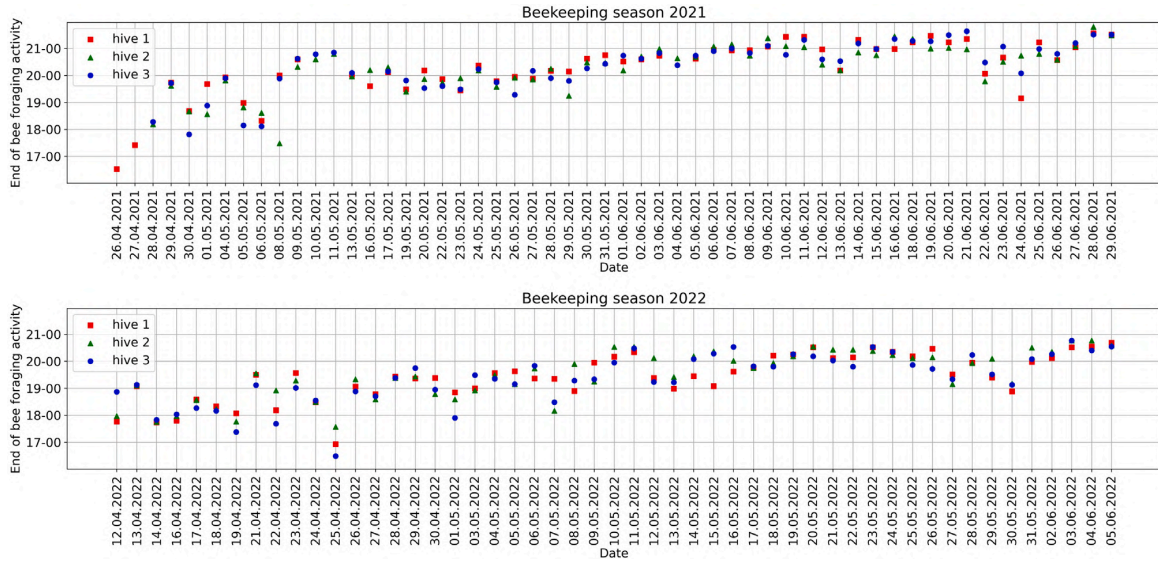


Fig. 3. Visualization of the end times of the bee foraging activity for the 2021 and 2022 beekeeping seasons.

from the skeleton coordinates obtained after skeletonization (Zhang and Suen, 1984) of the binary mask using linear regression. The study used the Mask R-CNN implementation from the detectron2 (Wu et al., 2019) library and the skeletonization algorithm implementation from the scikit-image (van der Walt et al., 2014) library.

### 2.5. Initial fitting and re-fitting of the regression model

Regression analysis techniques were used to predict the time remaining until the end of the foraging activity of the bees. The optimality of using a specific model depends on the character of the data, and for this reason we checked different regression models, and selected the best one based on the lowest prediction error (RMSE). The following models from the scikit-learn (Pedregosa et al., 2011) Python library were evaluated: *GradientBoostingRegressor* (GBR), *LinearRegression* (LR), *HuberRegressor* (HR), *BayesianRidge* (BR), *KernelRidge* (KR), *MLPRegressor* (MLP). The default parameter values for these models were used.

In addition, the 'FixedThresholdBaseline' (FTB) model was defined as the baseline. It is based on determining such an offset to the time to sunset that has the lowest prediction error. For the 'FixedThreshold-Baseline' model, features related to bee activity and weather conditions are not included.

### 2.6. Regression model initial fitting and re-fitting

An important step in the development of regression models is their initial fitting and eventual re-fitting. In our study, two types of fitting were considered.

The initial model fitting consisted of fitting the model on an initial training set that contains samples from days within a time window of length  $t_{train}$ .

Maintaining high accuracy of the model when the nature of the input data changes requires periodic model re-fitting. The problem under consideration is characterized by the ability to retrieve true (or semi-true) target values (represented by the remaining bee foraging activity time (RBFAT) values) with a delay, i.e., at the earliest time after the bees have finished their daily foraging activity. Annotation by an expert (type *true* target values) for each day and each hive involves the expert determining the time of the end of bee foraging activity based

on the analyzed image sequence. After receiving the time of the end of the bee foraging activity, chunks are sequentially annotated with the corresponding RBFAT value and added to the training set. The study also proposed methods for automatic determination of the end of the bee foraging activity (type *semi\_true* target values). The most intuitive method to determine the end of daily bee foraging activity is the time of observing the last bee located at the entrance of the hive (type *semi\_true\_raw\_last* target values).

After each day, the model is re-fitted on the current training set. The study considered the following options for modifying the training set:

- *fixed* - only initial fitting is performed, the training set is not modified, no model re-fitting occurs,
- *landmark* - initial fitting and re-fitting of the model is performed, the training set is continuously increased, no removal of older samples occurs,
- *sliding* - initial fitting and re-fitting of the model is performed, the training set remains similar in size and contains samples from days within a time window of length  $t_{train}$ , removal of samples outside the time window occurs.

In order to protect the model from semi-true target values that are determined with high error, a regularization mechanism characterized by the parameter  $\lambda$  is proposed. The regularization involves that the final RBFAT value  $Y_{new}$  consists of two components:  $Y_{semi\_true}$  - representing the determined semi-true RBFAT value, and  $Y_{pred}$  - representing the predicted RBFAT value using the old model. Finally, the final RBFAT value is calculated using the formula:

$$Y_{new} = \lambda Y_{semi\_true} + (1 - \lambda) Y_{pred} \quad (1)$$

For the initial training, only component  $Y_{semi\_true}$  is used. The  $\lambda$  parameter can take values from 0 to 1, and specifies the percentage importance of the  $Y_{semi\_true}$  component in the formation of  $Y_{new}$ . By using a linear combination of  $Y_{semi\_true}$  and  $Y_{pred}$  in the formula for  $Y_{new}$  and a range of  $(0; 1)$  for the  $\lambda$  parameter,  $Y_{new}$  is always within the interval  $(Y_{min}, Y_{max})$ , where  $Y_{min} = \min\{Y_{semi\_true}, Y_{pred}\}$  and  $Y_{max} = \max\{Y_{semi\_true}, Y_{pred}\}$ .

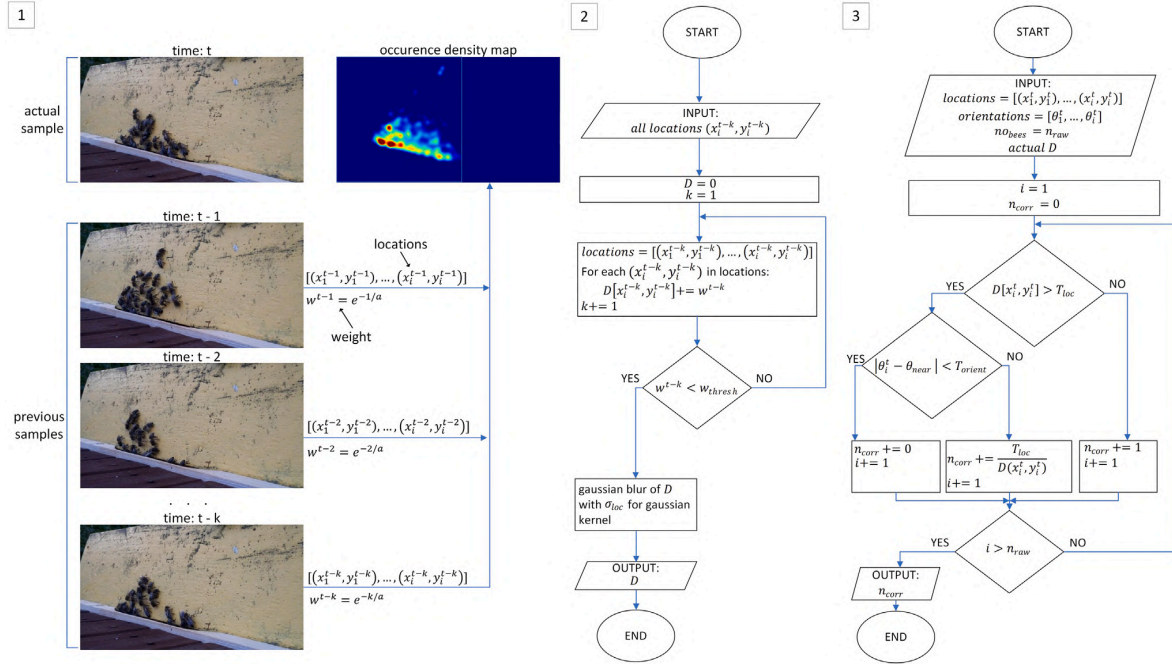


Fig. 4. The spatio-temporal correction: (1) subsequent samples taken for the calculation occurrence density map, (2) diagram of the calculation occurrence density map, (3) diagram of the calculation of the corrected number of bees.

### 2.7. Spatio-temporal correction

The presence of bees at the entrance of the hive is not synonymous with foraging activity. It is also possible to observe: (1) dead bees, (2) bees ventilating the hive, and (3) bees performing cleaning tasks, e.g. removing dead bees from the hive. These patterns can be detected by analyzing similarities in the location of bees in successively captured images.

In the context of the conducted research, the described phenomena make it difficult to correctly determine the end of the daily foraging activity of bees, because their occurrence can be incorrectly perceived as ongoing bee foraging. This problem is significant, especially when determining the automatic time of the end of bee foraging activity. The use of the naive semi-true target values determination strategy of treating the last detected bee as the end of bee foraging (*semi\_true\_raw\_last*) causes the automatically determined end time of bee foraging to be later than the true time (when other behavior patterns are present).

In order to reduce the error of the automatically determined time of the end of bee foraging activity, a spatio-temporal correction algorithm was proposed. It excludes or reduces the weight of bees, which are characterized by a similar location and orientation with respect to bees from previously captured images.

For each time step associated with the acquisition of a new image after prediction by the Mask R-CNN, a binary mask for each detected bee is obtained. For each binary mask, we compute the midpoint  $M_i(x_i, y_i)$  and the orientation  $\theta_i$ , thus obtaining the set of bee locations  $[(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)]$  and the set of bee orientations  $[\theta_1, \theta_2, \dots, \theta_n]$ , where  $n$  is the number of detected bees. The location of each bee is compared to the actual occurrence density map  $D$ , which is calculated based on the location of the detected bees in the previous image. If the location of a bee  $(x_i, y_i)$  indicates a point with a higher probability of occurrence ( $D(x_i, y_i) > T_{loc}$ ), its weight is reduced relative to the bees located in an area with a lower probability of occurrence. The contribution of a correlated bee to the total number of bees in the image is  $\frac{T_{loc}}{D(x_i, y_i)}$ , while an uncorrelated bee - 1. The orientation of the

correlated bee  $\theta_i$  is then compared to the orientation of the nearest bee  $\theta_{near}$  in the previous image. If  $|\theta_i - \theta_{near}| < T_{orient}$ , the detected bee should be considered as dead and is not taken into account when counting all the bees. Additionally, the nearest bee from the previous image, to be considered as dead, should be at a distance less than  $r_{orient}$ . The bee counting with spatio-temporal correction is summarized in Fig. 4.

After calculating the corrected number of bees for a given timestamp, the occurrence density map  $D$  is updated. The most relevant (having the highest weight) for the calculation of  $D$  are the locations of bees detected in recently captured images. The weights were calculated by taking into account the differences between the recording time of the current image and the recording time of the previous image  $t_{diff}$  using the following formula:

$$weight = e^{-t_{diff}/a} \tag{2}$$

The weights decrease exponentially as  $t_{diff}$  increases. A threshold of weights was set at  $w_{thresh}$  in order to eliminate insignificant instances concerning map estimation. After obtaining the raw form of the occurrence density map, a Gaussian blur of the map, characterized by the standard deviation for the Gaussian kernel  $\sigma_{loc}$ , was applied. The calculation of the occurrence density map is summarized in Fig. 4.

In this study,  $T_{orient} = 10^\circ$ ,  $r_{orient} = 20$  pixel, and  $w_{thresh} = 0.01$  were assumed, and the parameter values  $a$ ,  $\sigma_{loc}$ ,  $T_{loc}$  were fine-tuned. The parameter values used for the fine-tuning are listed in Table 1.

Two types of semi-true target values, determined after spatio-temporal correction, were defined. The *semi\_true\_STC\_last* approach is connected with the last bee that is observed after removing the bees with a high location and orientation similarity (most probably dead bees), and the *semi\_true\_STC\_no\_corr* approach is connected with the last bee for which  $D(x_i, y_i) < T_{loc}$ , allowing bees with a high location similarity (e.g., bees that ventilate the hive) to be excluded.

### 2.8. Fine-tuning of the methods' parameters

A summary of the parameters for the proposed methods is shown in Table 1. These parameters were fine-tuned in successive stages. In

**Table 1**  
Parameters for the proposed method.

Stage	Parameter	Description	Values
I	<i>model</i>	Machine learning model used for prediction	['GradientBoostingRegressor' (GBR), 'LinearRegression' (LR), 'HuberRegressor' (HR), 'BayesianRidge' (BR), 'KernelRidge' (BR), 'MLPRegressor' (MLP), 'FixedThresholdBaseline' (FTB)]
II	<i>t<sub>chunk</sub></i> <i>t<sub>bin</sub></i>	Size of time window used for prediction Size of bin used for features accumulation	[30, 60, 90, 150, 240] min [1, 2, 5, 10, 30] min
III	<i>use<sub>temp</sub></i> <i>use<sub>hum</sub></i> <i>use<sub>press</sub></i>	Flags indicating the use of temperature, humidity, pressure values during inference	[000, 100, 110, 101, 111]
IV	<i>t<sub>train</sub></i> <i>w<sub>type</sub></i>	Size of time window used to re-fit model Type of window used to re-fit model	[3, 5, 7, 10, 14] days ['fixed', 'landmark', 'sliding']
V	<i>a</i> <i>σ<sub>loc</sub></i> <i>T<sub>loc</sub></i>	Constant used for calculating weights for samples Standard deviation for Gaussian kernel used in blurring of occurrence density map Threshold to assess whether bee location similarity is significant	[5, 10, 30, 60] [50, 100, 400] pixels [0, 1, 2]
VI	<i>target_ values_ type</i>	Type of target values used to re-fit model	['true', 'semi_true_raw_last', 'semi_true_STC_last', 'semi_true_STC_no_corr']
VII	<i>λ</i>	Regularization coefficient that determines contribution of determined semi-true RBFAT value to new RBFAT value	[0, 0.1, ..., 0.9, 1.0]

**Table 2**  
Settings of subsequent parameter fine-tuning stages for the proposed methods.

Stage	<i>model</i>	<i>t<sub>chunk</sub></i>	<i>t<sub>bin</sub></i>	<i>use<sub>temp</sub></i>	<i>use<sub>hum</sub></i>	<i>use<sub>press</sub></i>	<i>t<sub>train</sub></i>	<i>w<sub>type</sub></i>	<i>STC</i>	<i>a</i>	<i>σ<sub>loc</sub></i>	<i>T<sub>loc</sub></i>	<i>target values</i>	<i>λ</i>
I	var	240	10	1	1	1	7	slid.	no	–	–	–	true	1
II	opt	var	var	1	1	1	7	slid.	no	–	–	–	true	1
III	opt	opt	opt	var	var	var	7	slid.	no	–	–	–	true	1
IV	opt	opt	opt	opt	opt	opt	var	var	no	–	–	–	true	1
V	opt	opt	opt	opt	opt	opt	opt	opt	yes	var	var	var	semi-true	1
VI	opt	opt	opt	opt	opt	opt	opt	opt	yes	opt	opt	opt	var	1
VII	opt	opt	opt	opt	opt	opt	opt	opt	yes	opt	opt	opt	opt	var

each stage, some of the parameters were fixed (suboptimal), while some were variable, taking the values given in Table 1. The settings for the subsequent fine-tuning stages are shown in Table 2. Only data from the 2021 season were used for fine-tuning of the parameters. The selected optimal parameters were used for validating the method based on data from the 2022 season.

### 2.9. Evaluation

The evaluation consisted of comparing the true time remaining until the end of bee foraging activity  $\Delta t_{true}$  with the predicted time  $\Delta t_{pred}$ . The RMSE metric was used to determine the prediction error and was calculated according to the formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\Delta t_{true}^i - \Delta t_{pred}^i)^2}{n}} \quad (3)$$

where  $\Delta t_{true}^i$ ,  $\Delta t_{pred}^i$  denote the true and predicted time remaining until the end of bee foraging activity for the *i*th chunk, and *n* is the number of chunks.

The referenced RMSE values for individual days considered data from 3 three stations, while the RMSE values for the entire 2021 and 2022 seasons considered data from all considered days in the particular season.

The second measure used to evaluate the models was the coefficient of determination  $R^2$ , which measures how well the model fits the data. We calculate the coefficient of determination  $R^2$  using the formula:

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^n (\Delta t_{true}^i - \Delta t_{pred}^i)^2}{\sum_{i=1}^n (\Delta t_{true}^i - \overline{\Delta t_{true}})^2} \quad (4)$$

where RSS, TSS denote the residual and total sum of squares, and  $\overline{\Delta t_{true}}$  - the average of the true time remaining until the end of bee foraging activity.

### 3. Results and discussion

This section addresses the following topics successively: (1) bee detection and segmentation using Mask R-CNN; (2) parameter fine-tuning of the proposed methods; (3) prediction results for consecutive days of the 2021 and 2022 beekeeping seasons (as averaged RMSE values over hives for the compared approaches); (4) intra-day prediction results for selected days (as absolute values of predicted time); (5) the calculation of occurrence density maps and their use for behavioral pattern detection; (6) a strategy to combine predictions from different hives; and (7) the adaptability of the proposed methods for stream processing.

The first step in implementing the proposed methods was to train the Mask R-CNN instance segmentation model using labeled data from the 2021 season. The obtained average precision value  $AP_{30} = 94.5$  on an independent test set for bee detection is satisfactory from the point of view of the addressed problem. The large diversity of the samples in the training set enabled the model to be robust to varying bee appearance and size, dense scenes, and overexposure. Exemplary results for inference by the Mask R-CNN model are presented in Fig. 5, which also shows the result of determining midpoints and orientations for the considered samples.

After the determination of the considered features (weather indices, number of bees for the corresponding times), parameter fine-tuning was carried out for the proposed method of predicting the time remaining to the end of the daily bee foraging activity. A summary of the results from selected stages is shown in Fig. 6. The bar heights in the figure represent the RMSE prediction error averaged over the hives and the days during the 2021 beekeeping season.

In stage I, which is related to selecting the regression model for prediction, it was noted that the machine learning models generally performed better than the FTB reference model. The GBR (GradientBoostingRegressor) model achieved the best results and was used





Fig. 5. Mask R-CNN inference and determination of midpoints (read points) and bee orientations (angle of inclination of the green sections), e.g. samples captured at: (1) 10 a.m., (2) 3 p.m..

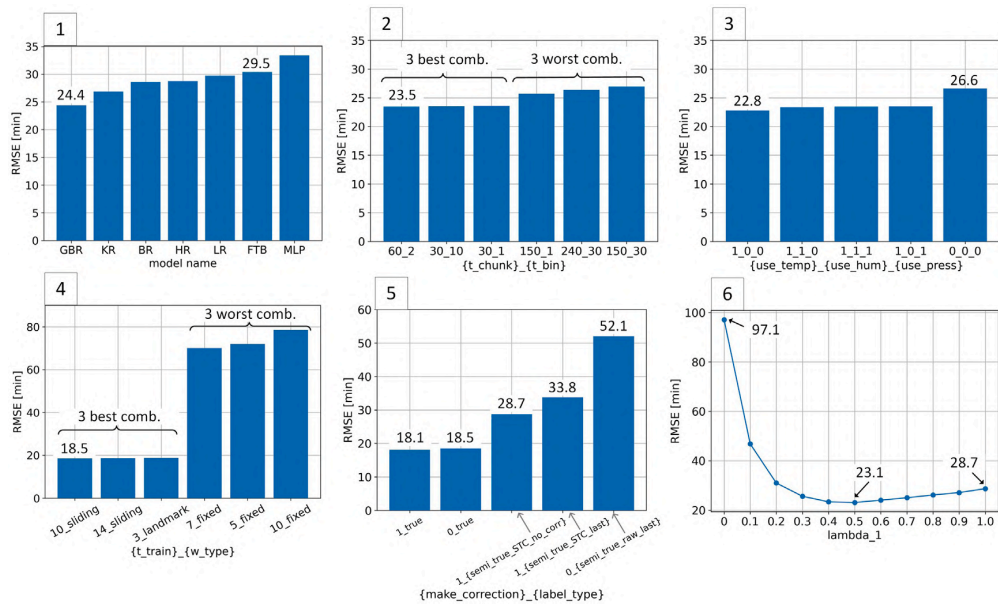


Fig. 6. Results of the subsequent parameter fine-tuning stages that are related to the selection of: (1) the regression model, (2) the size of the chunk and bin, (3) weather characteristics (temperature, humidity, pressure), (4) the settings for model re-fitting (type and length of the time window), (5) the type of target values used for re-fitting (with information about making a correction), (6) the regularization parameter with regards to the calculation of new target values.

in the subsequent fine-tuning stages. In stage II, the optimal chunk and bin size for pre-processing the time-series data was checked. No improvement in inference was observed when increasing the chunk size. Bee activity collected from the last 60 min was sufficient to obtain optimal results. The parameter values  $t_{chunk} = 60$  and  $t_{bin} = 2$  were used for the next stages. The analyses carried out in stage III, which are related to weather conditions, showed that temperature was the only important characteristic to use in the prediction. In stage IV, different settings were checked for the model's re-fitting. The best results were obtained with the "sliding" strategy, together with a training window size of  $t_{train} = 10$ . These results are significantly better than when no re-fitting is applied ("fixed" strategy), which proves that re-fitting the model during the season is necessary to maintain high model quality. The results also show that older samples included in the "landmark" strategy can be omitted when re-fitting without reducing model accuracy. After choosing the optimal parameters for the spatio-temporal correction in step V, the prediction error for different approaches for determining semi-true target values was compared in step VI. The use of spatio-temporal correction reduced the error by about 23 min when compared to the results for the *semi\_true\_STC\_no\_corr* and the naive *semi\_true\_raw\_last* approaches. This error was further reduced by

about 6 min ( $RMSE = 23.1$  min) after applying the regularization mechanism and when fine-tuning the  $\lambda$  parameter. Finally, the difference between the method based on true target values ( $RMSE = 18.5$  min) and the method based on semi-true target values after the fine-tuning ( $RMSE = 23.1$  min) was only about 5 min. This demonstrates the lack of rationale for expert annotation during the season, and the possibility of relying on automatically generated semi-true target values. A summary of the determined optimal values during fine-tuning can be found in Table 3.

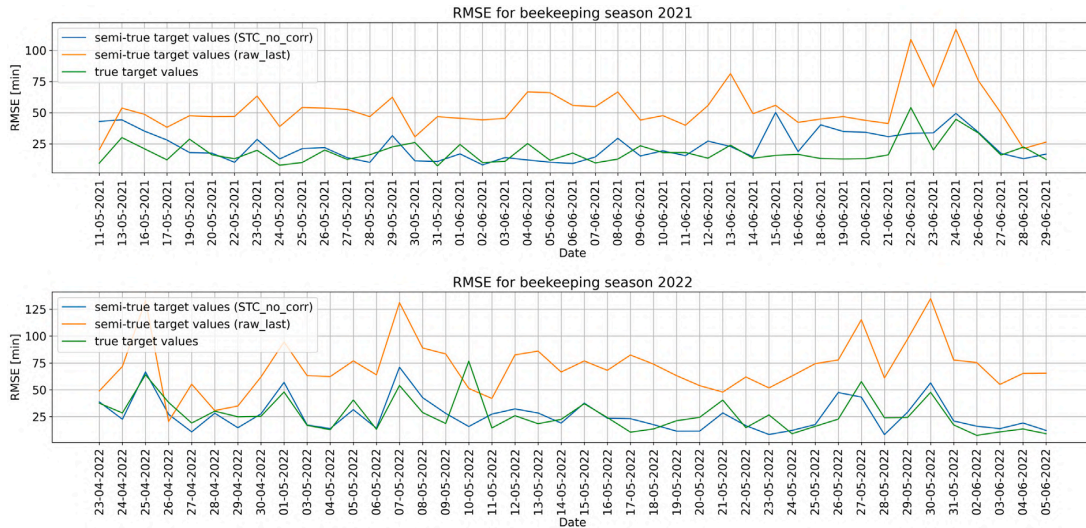
The obtained optimal parameters of the proposed methods were used for inference in the 2022 season. A summary of the results for the 2021 and 2022 seasons, showing the prediction error for consecutive observation days and different approaches, is presented in Fig. 7. The results for the most important approaches are also summarized in Table 4. The RMSE prediction error values for consecutive days (in Fig. 7) are the averaged RMSE values taken for each hive, whereas the RMSE prediction error values for the entire beekeeping season (in Table 4) are the averaged RMSE values taken for each hive and day of observation.

In the RMSE charts for the 2021 and 2022 seasons (Fig. 7), it can be seen that the *semi\_true\_STC\_no\_corr* approach (associated with

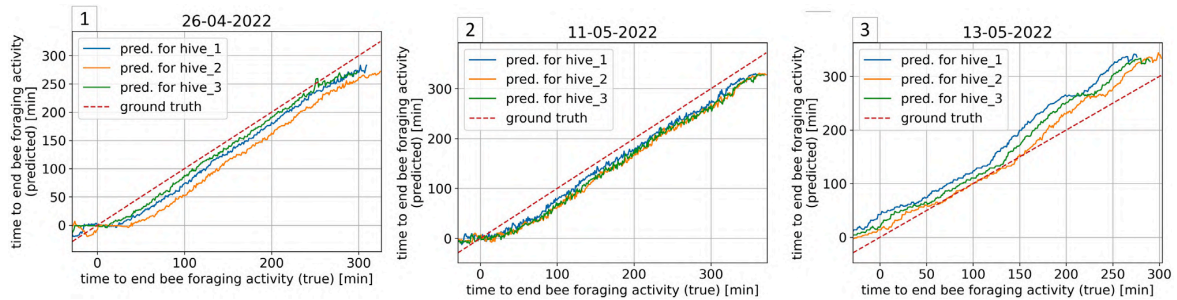


**Table 3**  
Optimal method parameters obtained after parameter fine-tuning.

model	$t_{chunk}$	$t_{bin}$	$use_{temp}$	$use_{hum}$	$use_{press}$	$t_{train}$	$w_{type}$	STC	$a$	$\sigma_{loc}$	$T_{loc}$	target values	$\lambda$
GBR	60	2	1	0	0	10	slid.	yes	60	100	0	semi_true (STC_no_corr)	0.5



**Fig. 7.** Summary results for the 2021 and 2022 beekeeping seasons, showing the prediction error RMSE (averaged over hives) for consecutive days of observation and selected approaches.



**Fig. 8.** Comparison of the absolute prediction values of the time to the end of bee foraging activity with the true values (ground truth) for the intra-day prediction on selected days.

**Table 4**  
Comparison of results for selected approaches after parameter fine-tuning for the 2021 and 2022 beekeeping seasons.

Season	target values	RMSE [min]	$R^2$
2021	true	18.5	0.958
	semi_true_raw_last	52.5	0.800
	semi_true_STC_no_corr (our)	23.1	0.930
2022	true	27.0	0.899
	semi_true_raw_last	71.2	0.660
	semi_true_STC_no_corr (our)	26.5	0.906

automatically generated semi-true target values) is able to maintain a similar RMSE error level as the *true* approach (associated with expert-determined target values throughout the beekeeping season under consideration). In order to better understand the prediction of the model, it is also useful to analyze the change in prediction as a function of the true time remaining until the end of bee foraging activity, as shown in Fig. 8. The figure shows the absolute values of the predicted time in comparison to the real time (ground truth).

In charts 1–3 in Fig. 8, a significant increase in the RMSE error (represented as the distance to the *ground truth* line) cannot be seen with an increasing true time remaining until the end of bee foraging activity. The reason for keeping the RMSE constant for large times is due to the use of the time-to-sunset feature  $\Delta t_{sunset}$ , which has regularization properties for predictions. Based on data from previous days, the model is able to initially estimate the end of bee foraging activity, which manifests itself by keeping the RMSE for large times approximately constant. The observation of reduced bee foraging activity at the end of the day results in a reduction of the RMSE error, which can be observed in the 0–60 min range for charts 1 and 2, and in the 0–120 range for chart 3.

The occurrence density maps used in this study not only made it possible to increase the veracity of the semi-true target values, but also allowed different patterns of bee behavior to be observed at the entrance to the hive. Selected bee behavior patterns are shown in Fig. 9.

Dead bees were observed on the occurrence density map as areas of a small area, and had a high probability of occurrence (2b in Fig. 9). Ventilation of the hive by bees could be observed as areas of a larger area, and had an increased probability close to the hive entrance (1b

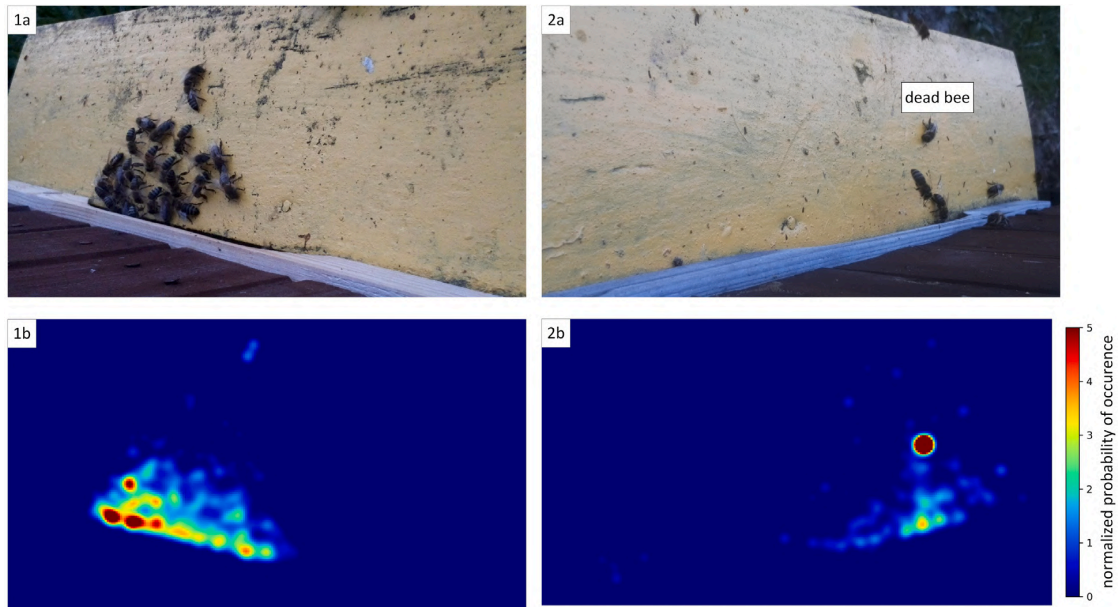


Fig. 9. Bee behavior patterns (1a, 2a) and corresponding occurrence density maps (1b, 2b): (1) hive ventilation by bees, (2) a dead bee.

in Fig. 9). Density maps provide much more information than the raw number of bees found at the hive entrance. They allow spatio-temporal information to be compressed and stored as a single 2D matrix, which enables it to be used successfully in stream data processing.

In our study, prediction was performed for each hive separately. In general, apiaries may consist of many hives, and it is necessary to consider in the final system how to combine predictions from many hives. The most intuitive strategy is the least favorable case, which is the hive for which the predicted remaining time is the longest.

The methods proposed in this work provide opportunities for their easy adaptation to stream processing. The most computationally expensive step in the presented approach is bee segmentation using Mask R-CNN. In order to increase the frequency of prediction, YOLO (Redmon et al., 2016; Jocher et al., 2020) models that are adapted for real-time prediction can be considered in the future for object detection. The ‘sliding’ approach will allow the accumulation of only the most recent data, and the occurrence density maps will compress the information about changes in the position of the bees in subsequent images.

#### 4. Conclusions

In our study, a method to predict the remaining time of bee foraging activity, taking into account bee activity, weather conditions, and time to sunset, was proposed.

Multistage parameter tuning of the proposed method enabled the optimal settings for minimizing prediction errors to be selected. GBR (Gradient Boosting Regressor) turned out to be the best regression model. Taking into account changes in temperature resulted in a decrease in the prediction error, with no effect on prediction errors being observed when humidity and pressure values were considered. The ‘sliding’ strategy was found to be the most appropriate when updating the training set.

The observation of significant changes in the nature of the data during the beekeeping season necessitated the proposal of mechanisms to maintain the quality of the model. The proposed mechanism of spatio-temporal correction, periodic model re-fitting, and regularization enabled significant error reduction for the methods based on

automatically generated semi-true target values, and also meant that it was reasonable to replace the true target values with semi-true target values. The evaluation results (RMSE = 23.1 min for 2021 and RMSE = 26.5 min for 2022) show that the proposed method of predicting the remaining time of honey bee foraging activity has great potential for application in a real-world scenario. The study also proves that it is possible to maintain high model quality throughout the season without the need for additional time-consuming annotation by an expert.

The proposed method to predict the remaining time of bee foraging can be a valuable component of a comprehensive advisory system for the planning of spraying and for exchanging information between farmers and beekeepers. The developed solution can help in the planning of advance spraying and transparently assess the end of bee foraging on a given day.

Future work should include: (1) analysis of in-field bee flight activity and the checking of the relationship of this activity to hive entry activity; (2) analysis of bee behavior patterns at the hive entrance using occurrence density maps or new feature representations; (3) the expanding of the dataset with more samples, especially data for different bee species and synthetic images; and (4) the development of a multifaceted system to prevent bee poisoning that is integrated with the methods proposed in this article.

#### CRedit authorship contribution statement

**Paweł Majewski:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **Piotr Lampa:** Conceptualization, Software, Data curation, Visualization, Writing – review & editing. **Robert Burduk:** Supervision, Writing – review & editing. **Jacek Reiner:** Conceptualization, Supervision, Writing – review & editing, Project administration, Funding acquisition.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

We would like to thank Henryk Majewski from H.T. Majewscy apiary (Łomnica-Folwark, Poland) for providing a data source and expert comments on the developed system. The research was supported by the Department of Laser Technology, Automation and Production Organization at the Faculty of Mechanical Engineering of Wrocław University of Science and Technology, Poland under the research subsidy for K62W10D07.

## References

- Alves, T.S., Pinto, M.A., Ventura, P., Neves, C.J., Biron, D.G., Junior, A.C., de Paula Filho, P.L., Rodrigues, P.J., 2020. Automatic detection and classification of honey bee comb cells using deep learning. *Comput. Electron. Agric.* 170, 105244.
- Andrijević, N., Urošević, V., Arsić, B., Herceg, D., Savić, B., 2022. IoT monitoring and prediction modeling of honeybee activity with alarm. *Electronics* 11 (5), 783.
- Bjerger, K., Frigaard, C.E., Mikkelsen, P.H.G., Nielsen, T.H., Misbiih, M., Kryger, P., 2019. A computer vision system to monitor the infestation level of Varroa destructor in a honeybee colony. *Comput. Electron. Agric.* 164, 104898.
- Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M., 2015. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons.
- Bozek, K., Hebert, L., Mikheyev, A.S., Stephens, G.J., 2018. Towards dense object tracking in a 2D honeybee hive. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 4185–4193.
- Bozek, K., Hebert, L., Portugal, Y., Mikheyev, A.S., Stephens, G.J., 2021. Markerless tracking of an entire honey bee colony. *Nature Commun.* 12 (1), 1–13.
- Campbell, J., Mummert, L., Sukthankar, R., 2008. Video monitoring of honey bee colonies at the hive entrance. In: *Visual Observation & Analysis of Animal & Insect Behavior, ICPR, Vol. 8*. pp. 1–4.
- Chan, J., Carrión, H., Mégret, R., Rivera, J., Giray, T., 2022. Honeybee Re-identification in Video: New datasets and impact of self-supervision. In: *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP*. SciTePress, INSTICC, pp. 517–525. <http://dx.doi.org/10.5220/0010843100003124>.
- Clarke, D., Robert, D., 2018. Predictive modelling of honey bee foraging activity using local weather conditions. *Apidologie* 49 (3), 386–396.
- Dembski, J., Szymański, J., 2020. Weighted clustering for bees detection on video images. In: *International Conference on Computational Science*. Springer, pp. 453–466.
- Gomes, P.A., Suhara, Y., Nunes-Silva, P., Costa, L., Arruda, H., Venturieri, G., Imperatriz-Fonseca, V.L., Pentland, A., Souza, P.d., Pessin, G., 2020. An amazon stingless bee foraging activity predicted using recurrent artificial neural networks and attribute selection. *Sci. Rep.* 10 (1), 1–12.
- He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 2961–2969.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 770–778.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9 (8), 1735–1780.
- Jocher, G., Nishimura, K., Mineeva, T., Vilariño, R., 2020. YOLOv5. <https://github.com/ultralytics/yolov5>.
- Lea, C., Vidal, R., Reiter, A., Hager, G.D., 2016. Temporal convolutional networks: A unified approach to action segmentation. In: *European Conference on Computer Vision*. Springer, pp. 47–54.
- Marstaller, J., Tausch, F., Stock, S., 2019. Deepbees-building and scaling convolutional neuronal nets for fast and large-scale visual monitoring of bee hives. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*.
- Ngo, T.N., Rustia, D.J.A., Yang, E.-C., Lin, T.-T., 2021a. Automated monitoring and analyses of honey bee pollen foraging behavior using a deep learning-based imaging system. *Comput. Electron. Agric.* 187, 106239.
- Ngo, T.-N., Rustia, D.J.A., Yang, E.-C., Lin, T.-T., 2021b. Honey bee colony population daily loss rate forecasting and an early warning method using temporal convolutional networks. *Sensors* 21 (11), 3900.
- Ngo, T.N., Wu, K.-C., Yang, E.-C., Lin, T.-T., 2019. A real-time imaging system for multiple honey bee tracking and activity monitoring. *Comput. Electron. Agric.* 163, 104841.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830.
- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 779–788.
- Rodriguez, I.F., Megret, R., Acuna, E., Agosto-Rivera, J.L., Giray, T., 2018. Recognition of pollen-bearing bees from video using convolutional neural network. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pp. 314–322.
- Rustia, D.J.A., Lu, C.-Y., Chao, J.-J., Wu, Y.-F., Chung, J.-Y., Hsu, J.-C., Lin, T.-T., 2021. Online semi-supervised learning applied to an automated insect pest monitoring system. *Biosyst. Eng.* 208, 28–44.
- Ryu, J.-S., Jung, J.-W., Jeong, C.-H., Choi, B.-J., Lee, M.-l., Kwon, H.W., 2021. Honeybee in-out monitoring system by object recognition and tracking from real-time webcams. *J. Apic.* 36 (4), 273–280.
- Semkiw, P., 2020. Sektor pszczelarski w Polsce w 2020 roku [Beekeeping sector in Poland in 2020]. Instytut Ogrodnictwa, Puławy.
- Skubida, P., 2007. Zatrucia pszczół, jako czynnik powodujący istotne straty w pszczelarstwie [bee poisoning as a factor causing significant losses in beekeeping]. *Pszczelarz Polski* (05), 10–12.
- de Souza, V.M., Silva, D.F., Batista, G.E., 2013. Classification of data streams applied to insect recognition: Initial results. In: *2013 Brazilian Conference on Intelligent Systems*. IEEE, pp. 76–81.
- Stojnić, V., Risojević, V., Pilipović, R., 2018. Detection of pollen bearing honey bees in hive entrance images. In: *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*. IEEE, pp. 1–4.
- Tashakkori, R., Hamza, A.S., Crawford, M.B., 2021. Beemon: An IoT-based beehive monitoring system. *Comput. Electron. Agric.* 190, 106427.
- Taylor, S.J., Letham, B., 2018. Forecasting at scale. *Amer. Statist.* 72 (1), 37–45.
- Wada, K., 2018. Labelme: Image polygonal annotation with python. <https://github.com/wkentaro/labelme>.
- van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Goullart, E., Yu, T., the scikit-image contributors, 2014. Scikit-image: image processing in Python. *PeerJ* 2, e453. <http://dx.doi.org/10.7717/peerj.453>, URL <https://doi.org/10.7717/peerj.453>.
- Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R., 2019. Detectron2. <https://github.com/facebookresearch/detectron2>.
- Zhang, T.Y., Suen, C.Y., 1984. A fast parallel algorithm for thinning digital patterns. *Commun. ACM* 27 (3), 236–239.

### 4.3 Monitoring the growth of insect larvae using a regression convolutional neural network and knowledge transfer

**Authors:** Paweł Majewski, Mariusz Mrzygłód, Piotr Lampa, Robert Burduk, and Jacek Reiner  
**Publication status:** published

**Type of publication:** journal paper

**Journal/Conference:** Engineering Applications of Artificial Intelligence (IF=8.0)

**MEiN points:** 140

**Lead Author:** Yes

**Corresponding Author:** Yes

**Percentage contribution:** 60%

**CRedit:** Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualisation, Writing – original draft preparation



Contents lists available at ScienceDirect

# Engineering Applications of Artificial Intelligence

journal homepage: [www.elsevier.com/locate/engappai](http://www.elsevier.com/locate/engappai)

## Monitoring the growth of insect larvae using a regression convolutional neural network and knowledge transfer

Paweł Majewski<sup>a,\*</sup>, Mariusz Mrzygłód<sup>b</sup>, Piotr Lampa<sup>b</sup>, Robert Burduk<sup>a</sup>, Jacek Reiner<sup>b</sup><sup>a</sup> Faculty of Information and Communication Technology, Wrocław University of Science and Technology, 27 Wybrzeże Wyspiańskiego st., 50-370 Wrocław, Poland<sup>b</sup> Faculty of Mechanical Engineering, Wrocław University of Science and Technology, 27 Wybrzeże Wyspiańskiego st., 50-370 Wrocław, Poland

### ARTICLE INFO

#### Keywords:

Growth monitoring  
Larvae phenotyping  
Tenebrio Molitor  
CNN-based regression  
Knowledge transfer  
Dense scenes

### ABSTRACT

Recently, there has been an increase in the popularity of breeding insect larvae (Tenebrio Molitor and Hermetia Illucens). Dimensioning larvae and observing their growth over time is a key component of monitoring insect larvae breeding. Due to the high number of larvae in the analysed images (dense scenes) and their overlap, determining the size distribution of larvae in real-time is a research challenge. In this work, we proposed an efficient method for determining the size distribution of larvae based on a regression convolutional neural network (RegCNN) and knowledge transfer. Larval width was chosen as the main measured larval parameter due to its ease of registration in dense scenes. The larval length  $L$  and its volume  $V$  were determined indirectly using determined regression models  $L(\text{width})$  and  $V(\text{width})$ . RegCNN training was performed using knowledge transfer to omit the time-consuming labelling of multiple images containing larvae at different growth stages. Training used quartiles (lower quartile, median, upper quartile) of larval widths determined using improved multistage larvae phenotyping based on classical computer vision methods and larvae segmentation model. Finally, our approach required labelling only a few images for calibration purposes. The study evaluated different RegCNN architectures: pre-trained on ImageNet (ResNet, EfficientNet) and custom with a reduced number of model parameters. The proposed method was validated for the distribution of larvae characterised by width quartiles taking values from 1.7 mm to 3.1 mm, corresponding to an average larval length of 16 mm to 28 mm. For the best evaluated model (ResNet18) in larval width estimation, we obtained RMSE = 0.131 mm (average RMSE = 1.12 mm for larval length estimation) and  $R^2 = 0.870$  (coefficient of determination) with an average inference time of 0.30 s/box. The best proposed custom architecture (TenebrioRegCNN\_v3) achieved slightly lower accuracy (RMSE = 0.134 mm,  $R^2 = 0.864$ ) with about five times lower inference time per image than ResNet18. The quantitative results confirmed the proposed method's potential to be applied in real breeding conditions.

### 1. Introduction

The breeding of insect larvae (mainly Tenebrio Molitor [TM] and Hermetia Illucens [HI]) is becoming an increasingly important part of the agri-food sector in Europe (Grau et al., 2017). The products obtained after breeding can be used for the production of protein feed (larvae) (Grau et al., 2017), bio-packaging (chitinous moult) (Priyadarshi and Rhim, 2020) and bio-fertilisers (frass) (Houben et al., 2020). Due to a decision by the European Commission in May 2021, mealworm larvae (Tenebrio Molitor) have also been authorised as a novel food, allowing them to be consumed by humans (EFSA Panel on Nutrition et al., 2021).

TM and HI insect breeding is characterised by large-scale production, which necessitates automation (Kröncke et al. (2020)). An

important element supporting the breeding of insect larvae is monitoring breeding to detect anomalies, which can be static (e.g. dead larvae, pests) or temporal (e.g. inconsistency of larval growth with the reference model of larval growth). Majewski et al. (2022) proposed a 3-module multipurpose system for monitoring Tenebrio Molitor breeding. The instance segmentation module (ISM) was responsible for the detection of the growth stages of the Tenebrio Molitor (larva, pupa, beetle) and anomalies in the form of dead larvae and the pest *Alphitobius diaperinus*. The semantic segmentation module (SSM) allowed the determination of the percentage coverage of the breeding box by chitinous moults, feed and frass. The larval phenotyping module (LPM) allowed the estimation of larval size parameters (length, volume) for individual larvae and the whole population. The authors emphasised that the LPM

\* Corresponding author.

E-mail address: [pawel.majewski@pwr.edu.pl](mailto:pawel.majewski@pwr.edu.pl) (P. Majewski).

<https://doi.org/10.1016/j.engappai.2023.107358>

Received 5 May 2023; Received in revised form 21 September 2023; Accepted 23 October 2023

Available online 2 November 2023

0952-1976/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).



module was the bottleneck of the entire system due to the relatively long inference time, which mainly consisted of: (1) segmentation of larvae using Mask R-CNN (He et al., 2017), (2) skeletonisation of larvae, (3) division of larvae into segments, (4) calculation of features for segments, and (5) classification of determined segments. LPM in the proposed form could not be applied under large-scale breeding conditions for real-time prediction. Adapting the larval phenotyping module to large-scale breeding conditions by significantly reducing inference time while maintaining adequate model accuracy was one of the main motivations for conducting the research described in this publication. Baur et al. (2022) proposed an indirect method of monitoring the growth of the *Tenebrio Molitor* larvae by recording changes in the size of larva segments. The solution included grayscale image thresholding, segmentation of the larvae using the watershed algorithm, classification of the extracted segments into four classes (good segments, medium segments, bad segments, and artefacts) using an artificial neural network, and dimensioning of the good segments. The solution developed may have problems at the early stages of the *Tenebrio Molitor* growth, when segments may not be well visible. Through the multistage nature of the solution discussed, it will also be difficult to adapt it to new data — an end-to-end solution would be better.

Apart from these two cited papers, to the best of our knowledge, researchers have not addressed the problem of monitoring the growth of insect larvae using computer vision and machine learning methods. However, the problem of size parameter estimation, weight estimation and animal growth monitoring has been addressed in the literature in the context of cattle (Wang et al., 2023), pigs (Bhoj et al., 2022), poultry (Nyalala et al., 2021) and fish (Li et al., 2020). One can find methods based on both classical computer vision methods (Weber et al., 2020; Pezzuolo et al., 2018) and newer approaches based on deep neural networks (Zhang et al., 2021; Gjergji et al., 2020; Cang et al., 2019). Classical methods are based on multistage image processing that incorporates algorithms based on specific domain knowledge — often in the form of a set of rules. The advantage of these methods is that they are transparent and easy to interpret the results obtained. On the other hand, these methods are difficult to adapt quickly when the nature of the data changes. Regression convolutional neural networks (RegCNN) are a very promising approach to the problems in question for estimating geometric quantities and weights, providing an end-to-end solution. The adaptation of RegCNN models to new data is based on repeated training for a new set of annotated data.

Konovalov et al. (2019) proposed an automatic method for estimating fish weights from 2D images. The first approach was based on (1) segmentation of fish from images using the LinkNet-34 model (Chaurasia and Culurciello, 2017) and (2) calculation of fish weights using a determined linear regression model for the relationship between weight and area of a binary mask. In the second approach, the segmentation part was omitted, and the weight was estimated directly from the image of the segmented fish using regression CNNs (LinkNet-34 adapted to the regression problem). Cang et al. (2019) developed a method for estimating pig weights from depth images of the back of pigs in top view. An extension of the Faster R-CNN model (Ren et al. (2015)) was proposed with a regression branch for determining the estimated pig weight. Training simultaneously minimised the loss associated with the recognition, localisation and weight estimation. Zhang et al. (2021) proposed a multiple output regression convolutional neural network (RegCNN) for estimating various size parameters and weight for pigs from depth images. The minimised mean squared error (MSE) loss during training considered body weight and five size parameters: shoulder width, shoulder height, hip width, hip height, and body length. RegCNN was developed by adapting pre-trained (on ImageNet) backbones to the regression task. DenseNet201 (Huang et al., 2017), ResNet152V2 (Yu et al., 2018), Xception (Chollet, 2017), and MobileNet V2 (Sandler et al., 2018) were used. In Gjergji et al. (2020), the weight of beef cattle was estimated from 2D images. The method assumed a combination of recurrent attention model (Mnih et al., 2014)

with a convolutional neural network based on the EfficientNet-B1 (Tan and Le, 2019) backbone. The use of an attention mechanism was argued to be more attentive to shape than texture.

From the work mentioned, it can be seen that there are significant differences between the phenotyping of insect larvae and that of other animals. In particular, it should be noted that the breeding of the mealworm takes place in boxes with a high density of individuals, resulting in dense scenes in the collected images. In the works cited, the strategy of one individual per image (e.g. one cattle in a sow stall) was used. Dense scenes result, on the one hand, in the difficulty of extracting uncovered individuals from the image and, on the other hand, in a significant number of individuals to be analysed. Performing the labelling of images containing dense scenes is extremely time-consuming. For this reason, when developing methods for the phenotyping of insect larvae, special attention should be paid to improving the development process of the methods (in particular reducing the time spent on the manual labelling of samples). It was not observed that this problem was addressed in the described works.

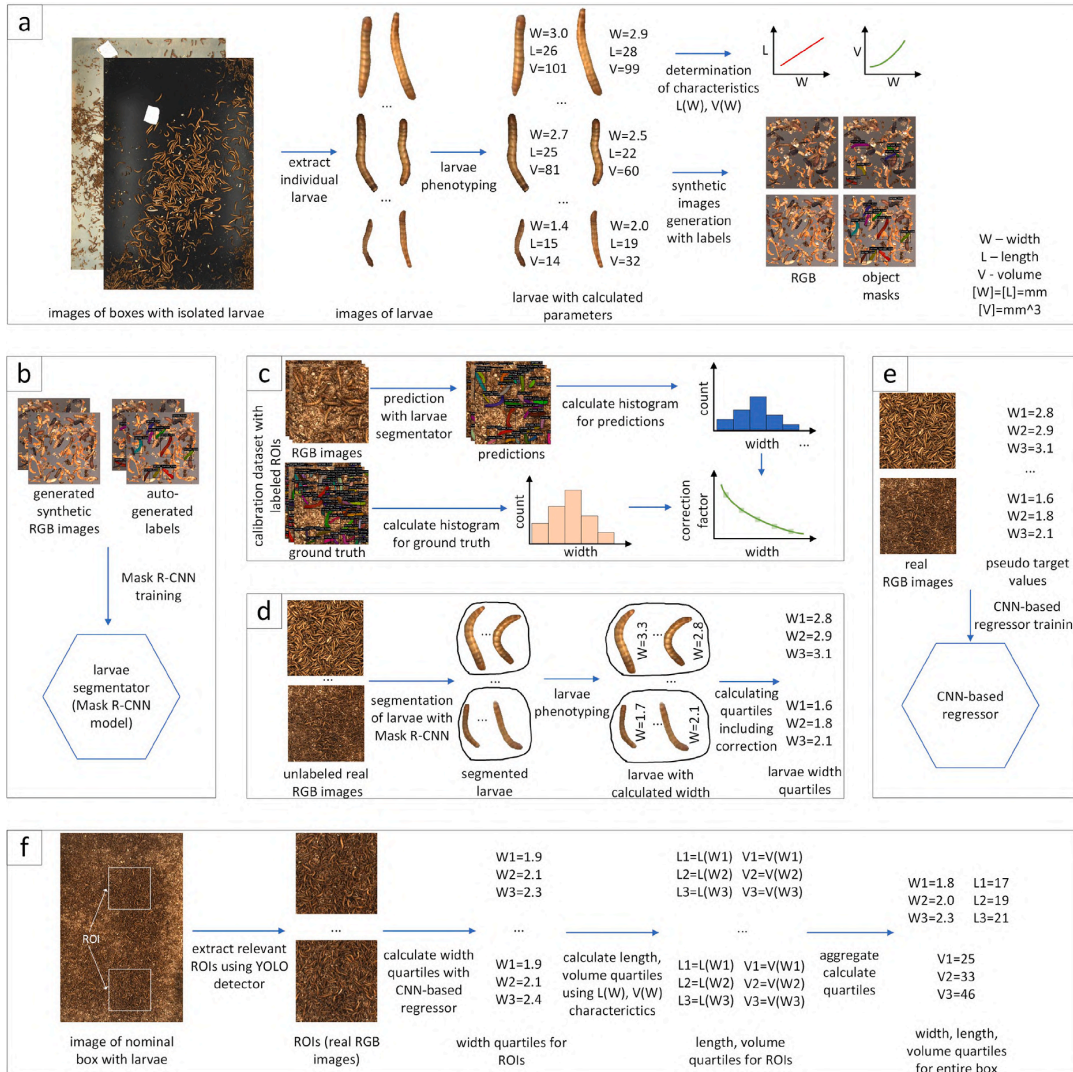
The problems of data augmentation and speeding up the annotation process are present in the literature in the context of other application problems. The topic of generating synthetic images using a simulation approach is worth noting. In Dolata et al. (2021), generated images representing dense scenes of potatoes were used to train a regression model for potato size distribution estimation, reducing the impact of object overlaps and perspective distortion on the results. Articles Abbas et al. (2021), Lu et al. (2019), on the other hand, have shown that the use of additional images generated using GAN (Generative Adversarial Network) (Goodfellow et al., 2014) in training deep convolutional neural network classifiers can contribute to the performance of the developed models. A rather interesting research direction in accelerating data annotation is pseudo-label-based self-training, which involves using the prediction of a weak model (trained on a relatively small number of manually labelled samples) as labels to train a subsequent model. Weak model inference is performed on large sets of unlabelled samples. This approach was used in Chaitanya et al. (2023) for the problem of semantic segmentation of medical images and in Shin et al. (2020) in the context of the domain adaptation problem for semantic urban scene understanding. The knowledge transfer mechanism has been used successfully to reduce the dimensionality of neural networks through teacher–student training (Sharma et al., 2018; Bergmann et al., 2020). In the context of the problem of analysing images of mealworms representing dense scenes addressed in this paper, it is important to consider the possibility of using an analogous mechanism to share knowledge between knowledge-based solutions (e.g. multistage phenotyping using computer vision) and end-to-end solutions, which could be provided by a regression convolutional neural network.

Considering the above, our proposed solution is a regression convolutional neural network (RegCNN) trained using a knowledge transfer mechanism between the RegCNN and an improved multistage larvae phenotyping method (based mostly on classical computer vision). Thanks to this approach and the use of automatically generated synthetic images at the step of developing an instance segmentation model of larvae in multistage phenotyping, it is possible to limit the labelling to only a few samples for calibration purposes. A machine vision system specifically designed for monitoring the breeding of insect larvae should also be recognised as an important element of the presented research.

## 2. Materials and methods

### 2.1. Definition of the problem

The problem under consideration is to propose an efficient method for phenotyping populations of insect larvae characterised primarily by (1) a small estimation error of size parameters (quartiles of width, length and volume of larvae), (2) a short processing time independent



**Fig. 1.** Scheme for the proposed solution: (a) determination of the linear regression models length(width) and volume(width), extraction of single larvae images, and generation of synthetic images, (b) development of a larvae segmentation model for multistage phenotyping, (c) correction factor determination for width quartiles calculation, (d) multistage phenotyping for selected samples, (e) training of a regression convolutional neural network using knowledge transfer, (f) prediction using a developed regression convolutional neural network.

of the number of larvae in the breeding box, and (3) ease of implementation in new breeding or adaptation to new breeding conditions. One larvae population is associated with one breeding box, where the larvae are located during breeding. The idea of the solution is presented in Fig. 1, and the steps of the proposed solution are discussed in detail in the following sections.

### 2.2. Data acquisition

Acquisition of images of the breeding boxes with insects was carried out in the conditions of industrial breeding, using a machine vision system placed on an automatic robot servicing the breeding. A real photo of the developed machine vision system is shown in Fig. 2. Images were acquired using the colour camera GOX-12401C (JAI, Denmark) with a resolution of 4096 x 3000 pixels and a lens of 12 mm

focal length. The camera was placed at a distance allowing for imaging of its entire surface. The camera's distance from the box's bottom surface was 487.5 mm. Those imaging conditions resulted in resolution of 0.143 mm/pixel. The acquisition area was illuminated with cool white LED strips that were triggered only for a short time of camera exposure to minimise the influence on the insects. The optical path was isolated by black covers to eliminate unwanted reflections. All optical elements were placed behind a glass sheet and enclosed in a housing to protect against dust occurring in breeding conditions.

The obtained raw images, before further processing, were subjected to the processes of compensation of shading resulting from insufficient lighting of the breeding box and removal of distortion produced by the lens used. Shading was compensated by means of a map of underexposed areas of the image, determined using a grey pattern. Distortion was removed using a chessboard pattern (Tsai, 1987).

**Table 1**  
Description of the defined datasets.

Dataset	Description
D1	10 images of size $4096 \times 3000$ containing isolated larvae at different growth stages. Some of the larvae in these images were manually annotated with a polygon annotation, resulting in 266 labelled larvae. Examples of images from the D1 dataset can be found in Fig. 1a.
D2	12 images of size $512 \times 512$ (extracted tiles from the whole images of size $4096 \times 3000$ ) containing larvae under real breeding conditions at different growth stages. Three subsets of samples of 4 images, each representing different growth stages of larvae characterised by the median length of the larvae, were extracted, namely: (18–23 mm), (23–27 mm), (27–32 mm). All larvae in the images in this dataset were manually annotated with polygon annotation type, resulting in a total of 1021 annotations. Examples of images from dataset D2 are in Fig. 1c.
D3	739 images of size $1024 \times 1024$ (extracted tiles from the whole images of size $4096 \times 3000$ ) containing larvae under real breeding conditions at different growth stages and with different larval densities. From this dataset, 489 images were selected for the training set (D3.TRAIN), 206 images for the first test set (D3.TEST.1) and 44 images for the second test set (D3.TEST.2). The D3.TEST.1 collection was used to validate the knowledge transfer between the multistage phenotyping method and the CNN regressor. The D3.TEST.2 collection was used to validate the accuracy of phenotyping with the CNN regressor. The target values in set D3.TEST.1 were the quartiles obtained during multistage phenotyping. The target values in set D3.TEST.2 were quartiles calculated from manually marked larvae in the images. The total number of annotations in the D3.TEST.2 set was 1977. Examples of images from the D3 dataset can be found in Fig. 1d and 1e, and in Fig. 3.

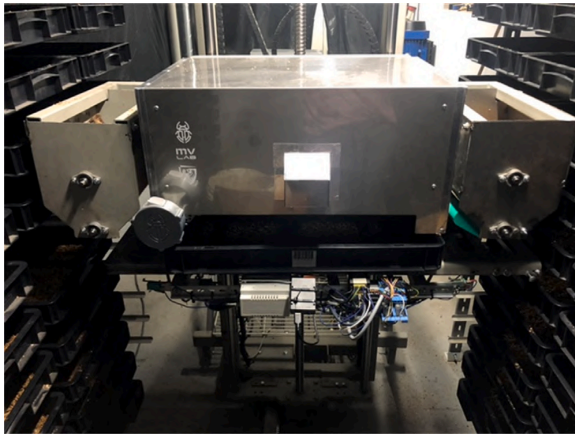


Fig. 2. The real photo of the developed machine vision system.

### 2.3. Data

To allow the development of the proposed models and their evaluation, three datasets were defined: (D1) a dataset containing the images used to determine the linear regression models length(width) and volume(width) and to extract images of individual larvae to generate synthetic images (Fig. 1a), (D2) a dataset containing the images used to determine the correction factor values and to evaluate the larvae segmentation model (Fig. 1c), (D3) a dataset containing the images used to train the regression convolutional neural network using knowledge transfer and to validate this model (Fig. 1d and Fig. 1e). Table 1 presents a detailed description of the defined datasets.

The images in datasets D2 and D3 represented square tiles with sizes of  $512 \times 512$  for D2 and  $1024 \times 1024$  for D3, respectively. Researchers commonly use the  $512 \times 512$  tile size for solving instance segmentation problems with Mask R-CNN (He et al., 2022; Shermeyer et al., 2021; Wu et al., 2020). In the case of the system proposed in this publication, this size was a compromise between longer computation time (large number of small-sized tiles) with probability of errors on the edges of the tiles and the ability to detect objects in dense scenes (limited number of proposals). For the phenotyping task with the CNN regressor, it was decided to increase the size to  $1024 \times 1024$  to increase the probability of finding the minimum number of larvae ( $n_{min}$ ) in a given area. The parameter  $n_{min}$  was introduced to avoid the borderline situation where phenotyping was done for an image containing no larvae or too few larvae for good-quality statistics. The value of the

parameter  $n_{min}$  was set to 10. The process of evaluating tile relevancy and the phenotyping procedure when the number of larvae is less than  $n_{min}$  were explained in more detail in the following sections of the publication.

The images from the D3 dataset were extracted at random locations from the raw images of mealworm boxes. These images were also manually checked for the presence of at least  $n_{min}$  larvae to ensure their relevancy. The D3.TRAIN and D3.TEST.1/D3.TEST.2 subsets were completely independent of each other. Independence was ensured at the level of the different breeding boxes. For the training set (D3.TRAIN), 16 boxes were selected, while 8 boxes were selected for the test set (D3.TEST.1 and D3.TEST.2). Examples of extracted tiles from the D3 dataset are shown in Fig. 3.

The images from datasets D2 and D3 were from a long-term feeding experiment conducted in October–November 2022. The growth of mealworm larvae in selected breeding boxes was monitored during this experiment.

### 2.4. Improved multistage phenotyping of larvae

A distinctive characteristic of the improved multistage phenotyping method described in this chapter is that it is mostly based on classical computer vision methods. Phenotyping according to this approach was used to (1) determine pseudo target values for training a CNN regressor using knowledge transfer (Fig. 1d) and (2) to determine the linear regression models length(width) and volume(width) (Fig. 1a). The term 'pseudo target values' refers to the values obtained using the improved multistage phenotyping method. The accuracy of larvae segmentation influenced the accuracy of 'pseudo target values'. The term 'true target values' will be used later in this publication to define the case where larvae were manually marked on the images.

Phenotyping can be carried out for a population of larvae (understood as all larvae contained in a single breeding box) or for individual larvae. Phenotyping a population of larvae involved determining the quantities that characterise the distribution of size parameters of larvae, i.e. the lower quartile (Q1), median (Q2) and upper quartile (Q3) for width, length and volume. To obtain size parameters for individual larvae, the first step was to extract individual larvae from the image using the instance segmentation model. In the next step, size parameters were determined for each larva using classical computer vision methods.

The basis for phenotyping individual larvae was a binary mask obtained after the segmentation of larvae, which precisely defines (pixel-wise) the area in the image where the selected larva was located. For further consideration, let us define two sets of pixels: the set of pixels included in the binary mask and the set of pixels contained in the contour of the binary mask. Phenotyping of individual larvae consisted of a couple of steps, which will be described in the next paragraphs. The



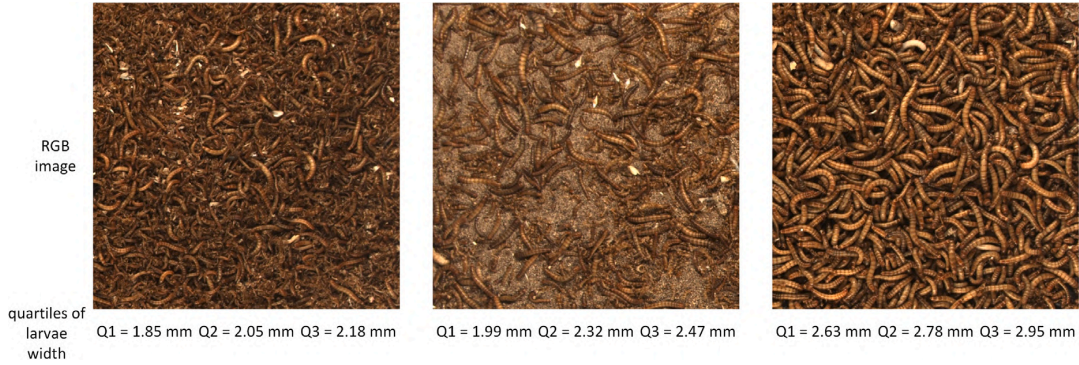


Fig. 3. Example images with corresponding quartiles of larvae width: lower quartile Q1, median Q2 and upper quartile Q3.

next steps of improved multistage phenotyping of individual larvae are shown in Fig. 4.

In the proposed improved multistage method for phenotyping single larvae, a customised method for determining the larval skeleton was used to remove the observed problems occurring with determining the skeleton using the built-in skeletonisation method from the scikit-learn library (Zhang and Suen, 1984; Pedregosa et al., 2011), as used in Majewski et al. (2022). The aforementioned standard skeletonisation method (Zhang and Suen, 1984) was based on sequentially removing contour pixels. This approach to skeletonisation resulted in a shorter skeleton, i.e. its ends were not at the points of the original contour. On the other hand, the method was sensitive to local noise, and the delineated skeleton was not smooth. For some samples, it was also possible to observe a problem with a significant change in orientation at the ends of the skeleton and a skeleton looping phenomenon.

The first step in phenotyping individual larvae was to select a random point inside the binary mask at a certain distance from the contour. This distance was defined as  $d_{contour}$ , and the selected point as the initial point. Then, we run a straight line through the initial point with such a slope that the line passed through as many pixels of the binary mask as possible (let us call such a line an auxiliary line). In the next step, we determined a straight line perpendicular to the auxiliary line and simultaneously passing through the initial point. Let us call the slope thus determined the initial slope of the straight line perpendicular to the skeleton. The points of intersection of the straight line perpendicular to the skeleton and the contour determined the section perpendicular to the skeleton. The midpoint of this section specified the skeleton point. The first skeleton point obtained was called the initial skeleton point. The determination of the initial point, the line perpendicular to the skeleton of the initial slope, and the initial point of the skeleton are shown in Fig. 4b.

Further skeleton points could be obtained by shifting the straight line perpendicular to the skeleton along the skeleton by  $d$ , which resulted in a change in the intercept value in the straight line equation by  $\Delta b = d / |\sin(\arctan(a_{skel}))|$ . Including in the formula the local orientation of the skeleton, expressed as the slope of  $a_{skel}$ , made it possible to obtain skeleton points approximately equidistant from each other by a value equal to the constant  $d$ . Moving a straight line perpendicular to the skeleton was done until the new line no longer had common points with the binary mask. In subsequent iterations of the proposed algorithm, the local orientation of the skeleton characterised by the slope  $a_{skel}$  was updated. A selected number of previously determined skeleton points  $n_{skel}$  was used to calculate  $a_{skel}$ . Determination of consecutive skeleton points by moving a line locally perpendicular to the skeleton is presented in Fig. 4c.

Obtaining all skeleton points required repeating the procedure for the two directions. The first direction was determined by shifting a straight line perpendicular to the skeleton by  $+ \Delta b$ , while the second

direction was determined by  $- \Delta b$ . The determination of the skeleton points for both directions is shown in Fig. 4c and 4d. After obtaining the set of points composing the skeleton  $S \in (x_1, y_1), \dots, (x_n, y_n)$  we can calculate the length of the larva as the sum of the lengths of the sections between consecutive points of the skeleton from the formula:

$$L = k \sum_{i=1}^{n-1} l(s_{i+1}, s_i) = k \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (1)$$

The coefficient  $k$  is a constant that converts pixels to millimetres.

The width of the larva was calculated as the median of the lengths of the sections perpendicular to the skeleton contained in the area of the binary mask. The corresponding points of the contour determined the sections' boundaries defining the larva's width. The sections determining the local width of the larva are highlighted in red in Fig. 4e.

The volume of the larva was determined from the formula proposed in Majewski et al. (2022), where it was assumed that the volume of the larva could be approximated by the sum of the volumes of cylinders of height  $l_i$  and diameter  $d_i$ , where  $l_i$  is the length of the selected section of the skeleton and  $d_i$  is the width of the larva at the chosen point of the skeleton. A correction factor  $c$  was also introduced in the formula, the value of which was determined experimentally. The formula for the volume of a single larva is as follows:

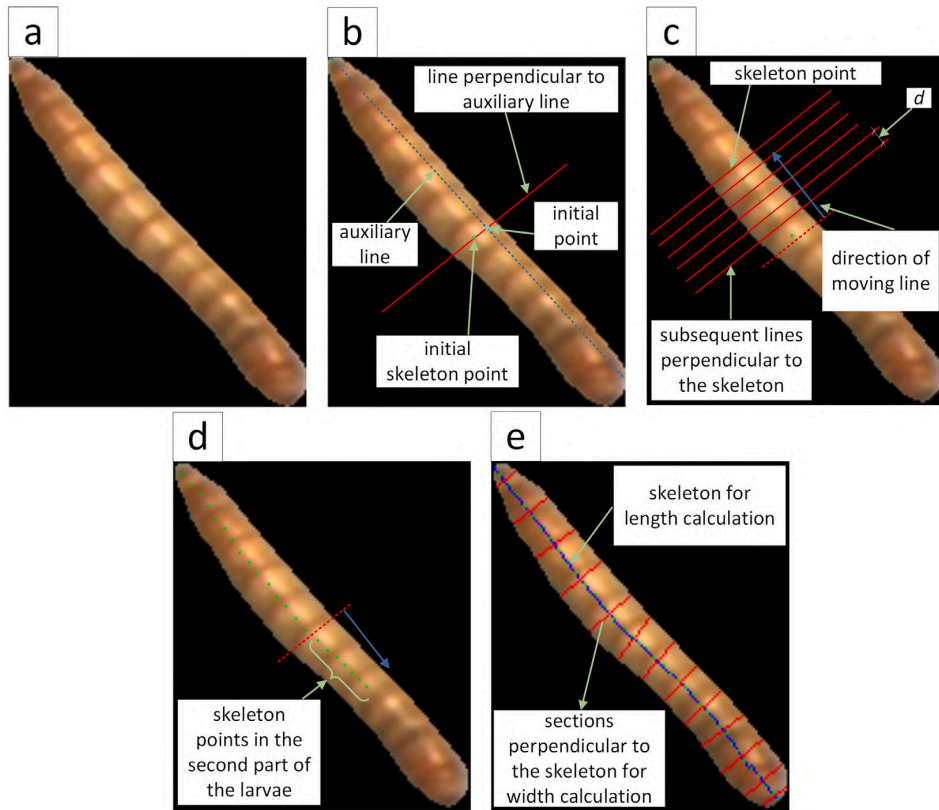
$$V = k^3 c \sum_{i=1}^{n-1} \frac{\pi}{4} d_i^2 l_i \quad (2)$$

The following values of constants were assumed in the study:  $d_{contour} = 5$  pix,  $d = 5$  pix,  $n_{skel} = 5$ ,  $k = 0.143$  mm/pix, and  $c = 0.58$ . With the chosen value of  $d_{contour}$ , there were no undesirable boundary phenomena and the determined initial points allowed the correct determination of the initial skeleton points. By increasing the  $d$  parameter, the skeleton determination time can be reduced; however, with too large values, information on the local orientation of the skeleton can be lost, forcing a compromise to be found for the value of this parameter. The parameter  $n_{skel}$  is responsible for the smoothing mechanism of the skeleton. With  $n_{skel} = 1$ , smoothing does not occur, and the method is sensitive to local noise. As with the  $d$  parameter, care must be taken with increasing the value of the  $n_{skel}$  parameter too much so that the effect of local orientation on the determined skeleton is appropriate. The parameter  $k$  is related to the developed machine vision system and was determined experimentally using a calibration standard of known dimensions. The value of the parameter  $c$  was determined experimentally in the article Majewski et al. (2022).

## 2.5. Determination of the linear regression models length (width) and volume (width)

The width of the larvae was chosen as the main and directly measured parameter of the larvae because of the ease of registration of this

larva with low curvature



larva with high curvature

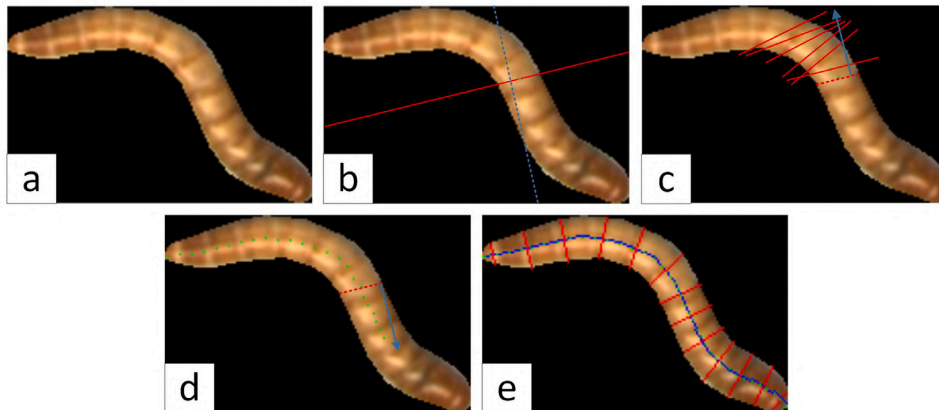


Fig. 4. Scheme for multistage phenotyping of single larvae: (a) image of a larva extracted from an image using a binary mask obtained after instance segmentation, (b) determination of the initial point, the line perpendicular to the skeleton with the initial slope and the initial point of the skeleton, (c) determination of successive skeleton points by moving a line locally perpendicular to the skeleton for the first part of the larvae, (d) determination of successive skeleton points for the second part of the larvae, (e) determination of the skeleton of the larva to calculate the length of the larva and a set of sections locally perpendicular to the skeleton to calculate the width of the larva.

dimension in dense scenes. However, from the breeder's point of view, the length parameter is easier to perceive. On the other hand, when determining the volumetric (or mass) gains of larvae during growth, larval volume is a more appropriate parameter. To allow indirect calculation of length and volume based on the measured width of

larvae, the linear regression models length (width) and volume (width) were determined.

Linear regression models length (width) and volume (width) were determined using larvae extracted from images from the D1 dataset, resulting in 266 points for the linear regression models determination.

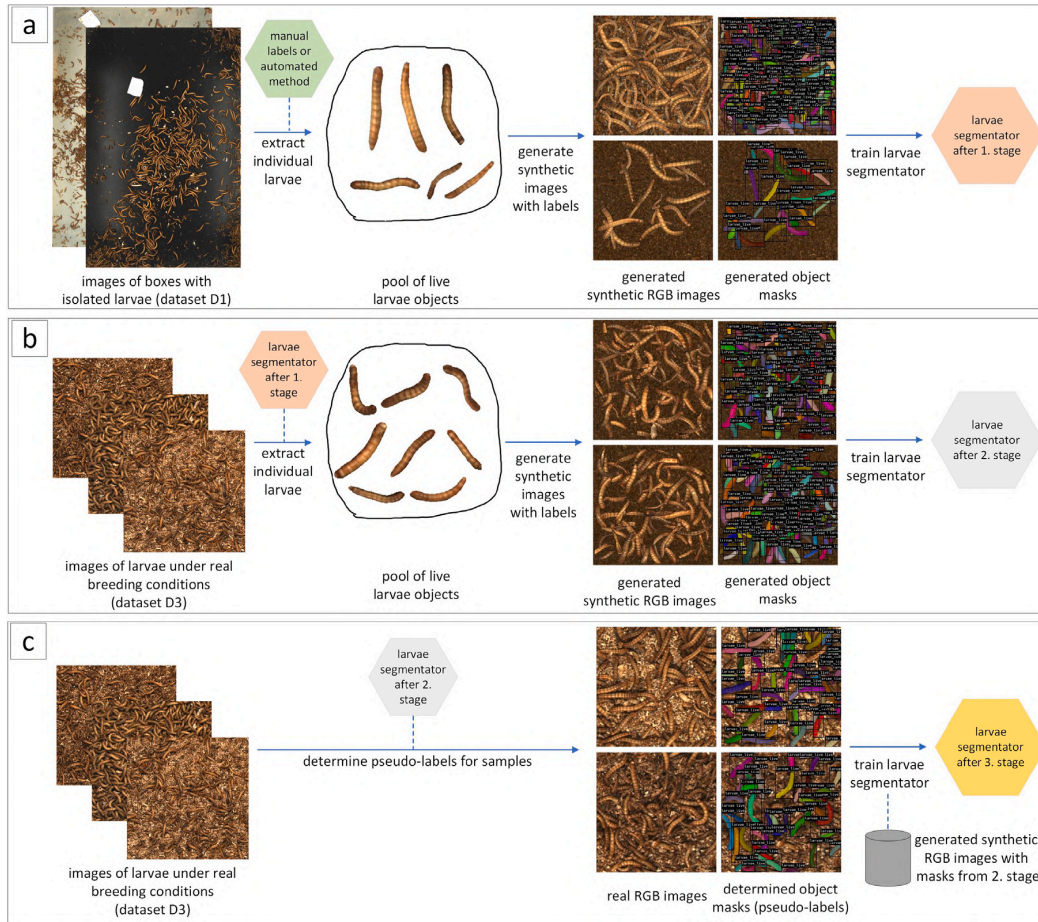


Fig. 5. The next steps in the development of the larvae segmentation model for multistage phenotyping: (a) training the model on generated synthetic images with object masks based on objects extracted from images from dataset D1, (b) training the model on generated synthetic images with object masks based on objects extracted from images from dataset D3, and (c) training the model on real images with object masks (pseudo labels) based on images from dataset D3.

The larvae were sized using the method described in Section 2.4. Based on the previously obtained points, a linear regression model was determined for the length(width) relationship and a degree-3 polynomial regression model for the volume (width) relationship. In determining the parameters of the regression model, it was assumed that the coefficients should have positive values. The scheme for determining the linear regression models length (width) and volume(width) is shown in Fig. 1a.

#### 2.6. Extraction of single larvae images and generation of synthetic images

The basic element of multistage phenotyping of larvae is their segmentation using the Mask R-CNN instance segmentation model. The development of such a model for the problem under consideration requires a set of labelled images. Labelling real images is very time-consuming due to the dense scenes. The solution to this problem is synthetic images with automatically generated labels. The synthetic image generation method uses previously prepared pools containing individual larvae images and involves randomly placing selected instances from the pool on the background image. The generation process is parameterised by the possible degree of coverage of neighbouring

instances and the number of instances to be placed in the image. The synthetic data generation approach for training the instance segmentation model was described in more detail in Majewski et al. (2022).

In the described study, the pool of instances consisted of larvae extracted from images from the D1 dataset — a total of 266 instances. Fig. 1a shows examples of synthetic images and automatically generated labels. The synthetic images were the basis for training the initial larvae segmentation model, which was improved in subsequent stages. The approach for developing the larvae segmentation model is described in the next Section 2.7.

#### 2.7. Development of a larvae segmentation model for multistage phenotyping

To accelerate the process of developing the larvae segmentation model, a original solution was proposed that required only a little user effort for labelling. The solution described in this chapter consists of three steps in which the efficiency of the larvae segmentation model was sequentially increased. The steps are shown in Fig. 5.

The basis of our solution was a pool of larvae instances extracted from images from the D1 dataset (266 instances in total). The larvae



images from the pool were used to generate synthetic data, as described in Section 2.6. The instance segmentation model obtained in step one (Fig. 5a) was trained on 200 generated synthetic images. The object pool used in step one consisted of 266 instances.

In step two (Fig. 5b), the inference was performed on the D3.TRAIN dataset using the model trained in step one, obtaining larval mask proposals for 489 images. Assuming a confidence score threshold of 50% and basic assumptions about the desired instance size, object pools were supplemented with new relevant larvae instances. The synthetic data generation process was repeated for the updated object pool, which consisted of about 65,000 larvae instances at this stage. The 1,000 generated synthetic images were used to train the instance segmentation model.

In step three (Fig. 5c), the inference was again performed on the D3.TRAIN dataset using the model trained in step two. This time, the extraction of individual larvae from the images was abandoned. The predictions of the previous model were treated as real labels, and the automatically labelled real images were used to train the model. In step three, the synthetic images from step two were also used to train the larvae segmentation model. The training set of samples contained both real and synthetic images.

The research used the Mask R-CNN (He et al., 2017) model with the ResNet101 backbone (He et al., 2016) developed for the instance segmentation problem. The Mask R-CNN implementation from the detectron2 library (Wu et al., 2019) was used. In each step, training was performed for 1600 epochs.

The number of synthetic images for the described experiment was chosen considering the number of objects in the object pool, i.e. each object had to appear at least once in the generated synthetic image, and the possibility of simulating different levels of larval density in the image. The selected number of epochs was sufficient to achieve adequate accuracy of the models, i.e. increasing the number of epochs did not noticeably contribute to improving the results on the validation set.

### 2.8. Correction factor determination for width quartiles calculation

The problem of detecting small objects by deep learning object detection (instance segmentation) models is often highlighted in the literature (Liu et al., 2021). In the context of our study, the possible different accuracy of the larvae segmentation model depending on the size of the instance can have a significant impact on the results obtained. To prevent the described problem, the calculation of a correction factor representing the values of the weights when calculating the larval width quartiles was proposed. The purpose of the correction factor was to determine the influence of a larva of a certain width on the value of the calculated quartile. It is expected that the values of the correction factor will decrease as the width increases. A diagram summarising the method for determining the correction factor as a function of larval width is shown in Fig. 1c.

The basis for calculating the correction factor was the D2 dataset described further in Table 1. The D2 dataset contained manually labelled larvae instances, allowing the calculation of the larval width histogram for ground truth. The dimensioning was carried out according to the method described in Section 2.4. On the other hand, a histogram was also determined for the predictions obtained after inference using the larvae segmentation model (Mask R-CNN model). The inference was performed for RGB images from the D2 dataset. Each bar in the determined histograms represented the number of larvae characterised by a width whose value is within a certain range. Let us denote  $h_{GT}^i$  as the  $i$ th histogram bar for ground truth and similarly  $h_p^i$  as the  $i$ th histogram bar for prediction. We define the correction factor for the  $i$ th width interval (bar) as  $c_i = h_{GT}^i/h_p^i$ . For the border case  $h_p^i = 0$ ,  $h_p^i = 1$  should be taken. The correction factor outside the considered width range assumes boundary values — the values of the correction factor for the first and last width range. Note that in the case  $h_{GT}^i = const$  decreasing  $h_p^i$  results in increasing  $c_i$ , which can be analysed as a higher weight when calculating width quartiles.

### 2.9. Multistage phenotyping for selected samples

The main aspect addressed in our publication is the knowledge transfer between improved multistage larvae phenotyping based on classical computer vision methods (described in Section 2.4) and a regression convolutional neural network (RegCNN). The knowledge transfer implied training RegCNN on values obtained from multistage phenotyping of larvae for samples from the D3.TRAIN dataset. For this purpose, the values of the lower quartile (Q1), median (Q2) and upper quartile (Q3) of larval width were determined for each sample from the D3.TRAIN dataset. When determining the quartiles values, the observations from the Section 2.8 section were considered, and correction weights were introduced when calculating the quartiles. This part of the proposed solution can be found in Fig. 1d.

### 2.10. Development and training of a regression convolutional neural network using knowledge transfer

The regression convolutional neural network (RegCNN) proposed in this study allowed the direct determination of values of larval width quartiles (Q1, median, Q3) without analysing individual larvae separately. The input to RegCNN was a  $800 \times 800$  RGB image. The output from RegCNN was the values of three quartiles of the larvae width. A scheme of this step of the proposed solution can be found in Fig. 1e.

During RegCNN training, the loss represented by the MSE (mean squared error) was minimised. For training RegCNN, images from the D3.TRAIN dataset were used. Before training, their size was reduced from  $1024 \times 1024$  to  $800 \times 800$ . Deep convolutional neural network architectures pre-trained on ImageNet (Krizhevsky et al., 2017) were evaluated: ResNet18, ResNet50, ResNet101 (He et al., 2016), EfficientNet-b0, EfficientNet-b4 (Tan and Le, 2019), MobileNetv2 (Sandler et al., 2018). Customised CNN architectures with reduced complexity have also been proposed.

For pre-trained models, fine-tuning was performed for all model weights (for both CNN and FC parts). In addition to the input layer in the FC, where the number of neurons depends on the type of backbone used, and an output layer containing three neurons (three quartiles of the larval width), three hidden layers were proposed. Depending on the number of neurons in the input layer, the numbers of neurons in the hidden layers were: for 512: [256, 128, 64], for 1024/1280: [512, 256, 128], for 1792/2048/4096: [1024, 512, 128]. A ReLU activation function was applied between successive layers in the FC. A scheme for the RegCNN with a pre-trained backbone structure used for the problem posed is shown in Fig. 6a.

For the custom RegCNN architecture, the model consisted of convolutional blocks and an FC block. The convolutional block consisted of a convolutional layer with a defined number and size of convolutional filters, ReLU activation and an average pooling layer with  $kernel\_size=2$  and  $stride=2$ . The FC block was defined similarly to the pre-trained case: (1) 3 hidden layers with an input and output layer, (2) the number of neurons in the hidden layers depending on the number of neurons in the input layer and (3) the ReLU activation function between successive layers in the FC block. A schematic of the custom RegCNN architecture used for the problem posed is shown in Fig. 6b. Additionally, in Table 2, the structure of the evaluated architectures is described.

RegCNN training was performed with the following parameter settings:  $learning\_rate = 0.001$ ,  $num\_epochs = 200$ ,  $loss = 'MSELoss'$ ,  $optimizer = 'Adam'$ . Before training, pixel intensity values in the R, G, B channels were standardised according to each channel's recommended values (based on ImageNet). The 'batch size' parameter for training was set separately for each model, considering the capabilities of the GPU hardware used. Training regressors above a chosen number of epochs no longer contributed considerably to improving the value of metrics on the validation set.

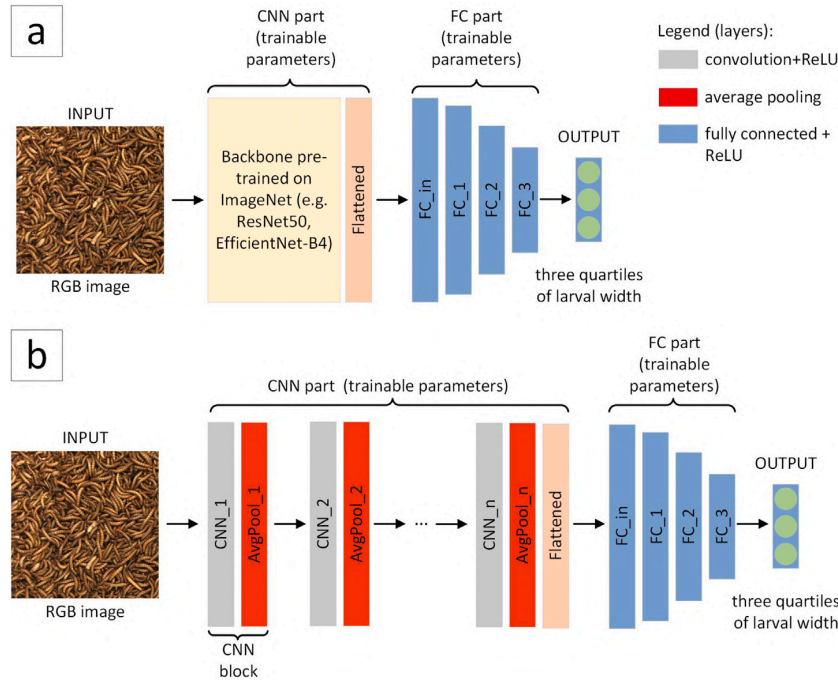


Fig. 6. Schematic structure of a regression convolutional neural network (RegCNN) based on: (a) pre-trained backbone on ImageNet, (b) custom architecture.

Table 2

CNN-based regressor structure for pre-trained backbones and evaluated custom architectures.

Model name	Model type	Params	No. CNN filters	Kernel	FC block structure
ResNet18	pretrained	11.3M	–	–	[512, 256, 128, 64, 3]
ResNet50	pretrained	26.2M	–	–	[2048, 1024, 512, 128, 3]
ResNet101	pretrained	45.2M	–	–	[2048, 1024, 512, 128, 3]
MobileNetV2	pretrained	3.0M	–	–	[1280, 512, 256, 128, 3]
EfficientNetB0	pretrained	4.8M	–	–	[1280, 512, 256, 128, 3]
EfficientNetB4	pretrained	20.0M	–	–	[1792, 1024, 512, 128, 3]
TenebrioRegCNN_v1	own archit.	5.2M	[16, 16, 32, 32, 64, 128, 256]	3	[4096, 1024, 512, 128, 3]
TenebrioRegCNN_v2	own archit.	1.8M	[16, 16, 32, 32, 64, 128, 256]	5	[1024, 512, 256, 128, 3]
TenebrioRegCNN_v3	own archit.	5.4M	[16, 32, 64, 64, 128, 128, 256]	3	[4096, 1024, 512, 128, 3]
TenebrioRegCNN_v4	own archit.	2.3M	[16, 32, 64, 64, 128, 128, 256]	5	[1024, 512, 256, 128, 3]
TenebrioRegCNN_v5	own archit.	2.8M	[16, 16, 32, 32, 64, 64, 128]	3	[2048, 1024, 512, 128, 3]
TenebrioRegCNN_v6	own archit.	0.6M	[16, 16, 32, 32, 64, 64, 128]	5	[512, 256, 128, 64, 3]

### 2.11. Prediction using a developed regression convolutional neural network

The breeding of the mealworm takes place in breeding boxes. The image of the entire breeding box was  $4096 \times 3000$ . The RegCNN described in Section 2.10 was based on  $1024 \times 1024$  tiles. To perform phenotyping for the entire breeding box, relevant regions of interest (ROIs) must be proposed in the image, where phenotyping with RegCNN will then be performed. The idea of prediction using RegCNN is presented in Fig. 1f.

Subsequent proposed ROIs were obtained using a sliding window approach with a sliding step of 1024, resulting in 12 candidates for relevant ROIs. ROIs were considered suitable for phenotyping if a minimum number of larvae  $n_{min}$  were present. The ROIs' relevance was assessed by a previously trained YOLOv5m (Jocher et al., 2020) object detection model characterised by fast inference time. The training of the YOLOv5m model was performed on the same samples as the Mask R-CNN model, changing only the form of the labels from polygons to bounding boxes. ROI relevance assessment was introduced into the prediction to avoid erroneous phenotyping results when a very small number of larvae are visible, which happens quite often in the early stages of breeding when larvae hide in the substrate.  $n_{min} = 10$  and

a confidence score threshold of 80% (for larvae detection) in the ROI relevance assessment was assumed, effectively eliminating cases of phenotyping with RegCNN for ROIs without larvae. If problems with false-positive errors in larvae detection occur when analysing images from new sources, consideration should be given to increasing the values of these parameters and re-training with objects falsely detected as larvae.

After phenotyping for the relevant ROIs, larval width quartile values were obtained for each ROI. Using the determined larval width quartile values and the linear regression models length(width) and volume(width), the quartile values for length and volume were determined indirectly. The final component of the prediction for the entire breeding box was aggregating the results obtained from the relevant ROIs. The quartile values for the whole breeding box were obtained by calculating the median of the quartile values determined for the individual ROIs.

### 2.12. Evaluation

The research evaluated the proposed methods using standard metrics.

### 2.12.1. Evaluation of larvae segmentation models

For larvae segmentation models, AP50 metric values were determined for three defined size distributions in the D2 dataset. The AP50 metric is the average precision at the intersection over union (IoU) 50%. The value of the AP50 metric was calculated as the area under the precision–recall curve after appropriate interpolation of the curve points. A more detailed metric explanation can be found in [Padilla et al. \(2021\)](#). In addition, the F1-score metric for the optimal working point was also determined.

The process of 3-step development of larvae segmentation models was repeated 5 times. At each of these three stages, the model from the best epoch was selected based on the calculated metrics on the validation set. The referenced results in the paper were the metrics calculated on the test set. The validation set and test set for this experiment consisted of randomly selected samples from the D2 dataset: 3 images represented the validation set, and 9 images the test set. Independence between the validation set and the test set was maintained. Results in the publication were referenced as averaged AP50 and F1-score values over repeats, given with standard deviation.

### 2.12.2. Evaluation of regression convolutional neural networks

The following metrics were used to evaluate methods for determining larval width quartiles: RMSE (root mean squared error), coefficient of determination  $R^2$ , and Pearson correlation coefficient  $r$ . Let us assume that for the  $i$ th image (one sample), the true values of the quartiles are:  $g_1^i, g_2^i, g_3^i$  and the predicted values of the quartiles:  $p_1^i, p_2^i, p_3^i$  respectively. In this situation, we calculate the RMSE from the formula:

$$RMSE = \sqrt{\frac{1}{3n_{sample}} \sum_{i=1}^{n_{sample}} \sum_{j=1}^3 (g_j^i - p_j^i)^2} \quad (3)$$

where  $n_{sample}$  is the number of samples.

We calculate the coefficient of determination ( $R^2$ ) from the formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n_{sample}} \sum_{j=1}^3 (g_j^i - p_j^i)^2}{\sum_{i=1}^{n_{sample}} \sum_{j=1}^3 (g_j^i - \bar{g})^2} \quad (4)$$

where  $\bar{g}$  - average true quartile value (value averaged over all samples and over all 3 types of quartiles).

We calculate the Pearson correlation coefficient ( $r$ ) from the formula:

$$r = \frac{\sum_{i=1}^{n_{sample}} \sum_{j=1}^3 (g_j^i - \bar{g})(p_j^i - \bar{p})}{\sqrt{\sum_{i=1}^{n_{sample}} \sum_{j=1}^3 (g_j^i - \bar{g})^2} \sqrt{\sum_{i=1}^{n_{sample}} \sum_{j=1}^3 (p_j^i - \bar{p})^2}} \quad (5)$$

where  $\bar{p}$  - average predicted quartile value.

For the evaluation of methods to determine larval width quartiles, datasets D3.TEST.1 and D3.TEST.2 were used. Dataset D3.TEST.1 contained pseudo target values for regression (obtained after multistage phenotyping) and was used to validate knowledge transfer between the multistage phenotyping method and the RegCNN. Dataset D3.TEST.2 contained true target values for the regression (obtained from manual annotations) and was used to validate phenotyping accuracy with the RegCNN.

For each evaluated architecture of the regression convolutional neural network, k-fold cross-validation was performed at  $k=5$ . The D3.TRAIN training set was divided into 5 approximately equal parts. In each of the five iterations, training was performed on 4 different parts (each part exactly once acting as a validation set). The results reported in the publication for sets D3.TEST.1 and D3.TEST.2 were the averaged values of RMSE,  $R^2$  and  $r$  metrics over five iterations.

### 2.12.3. Determination of processing time

For the processing time determination, hardware with the following specifications was used: GeForce RTX 2060 SUPER 8 GB (GPU) and AMD Ryzen 7 1700 3 GHz (CPU).

The study determined processing times for inference for individual images (inference time per image and throughput metrics) and for the whole pipeline (inference time per box metric). The term 'image' in the case of the 'inference time per image' metric refers to inference for a single tile of size  $1024 \times 1024$  ( $800 \times 800$  after resizing). The whole pipeline analysis determined the total time to analyse  $4096 \times 3000$  box images, which are divided into  $1024 \times 1024$  tiles. When analysing time for the whole pipeline, the total time also included image pre-processing time, larvae segmentation time with Mask R-CNN or ROI relevance assessment time with YOLOv5m, larvae phenotyping time (using RegCNN or multistage phenotyping) and post-processing time.

To calculate the 'inference time per image' metric, the prediction was repeated 1000 times on a single tile. The results were averaged and were referenced in the article with the standard deviation. For the calculation of the throughput (images per second) metric, inference was performed in batch mode, first determining the maximum possible batch size for a specific model, taking into account hardware limitations. Prediction for a specific batch was repeated 100 times in 5 iterations. Throughput values were averaged and were given in the article with standard deviation.

To calculate the total processing time when analysing the whole pipeline, 30 box images characterised by different sizes and densities of larvae were chosen. The prediction for each box image was repeated 5 times. As a result of this analysis, the paper reported the averaged inference time ( $t_{mean}$ ) with standard deviation ( $t_{std}$ ) and the minimum and maximum inference times ( $t_{min}, t_{max}$ ). The best architecture obtained according to the RMSE metric was selected for prediction using the CNN regressor.

## 3. Results and discussion

The results obtained were reported in the following order: (1) phenotyping results for individual larvae using the improved multistage larvae phenotyping method, (2) obtained linear regression models length(width) and volume(width), (3) evaluation of larvae segmentation models for different size sets of larvae, (4) dependence of the correction factor on larval width and justification of the validity of the correction, (5) evaluation and parameter fine-tuning for a regression convolutional neural network (RegCNN) (custom architecture and pre-trained architecture), (6) inference time analysis of the whole pipeline in the proposed solution, and (7) change of larvae size parameters in an example feeding experiment.

### 3.1. Phenotyping results for individual larvae using the improved multistage larvae phenotyping method

Phenotyping results for chosen images of individual larvae using the improved multistage larvae phenotyping method are shown in [Fig. 7](#). Larvae characterised by different sizes and curvature were selected for validation. Samples a-j show larvae visible in full, while samples k-o show larvae partially occluded (fragments of larvae).

The determined values of length (L) and volume (V) for the k - o samples were intentionally placed in brackets, as in the final solution, these parameters were determined indirectly using the determined linear regression models length(width) and volume(width), as described in the next section. In [Fig. 7](#), we can observe the high performance of the phenotyping method and its robustness when phenotyping small larvae and larvae with high curvature. With the proposed skeletonisation method, it was possible to obtain smoothed skeletons ending at the contour of the larvae and to avoid the problems indicated with the [Zhang and Suen \(1984\)](#) method, i.e. sudden changes in skeleton orientation at the ends and the looping phenomenon. Shorter red

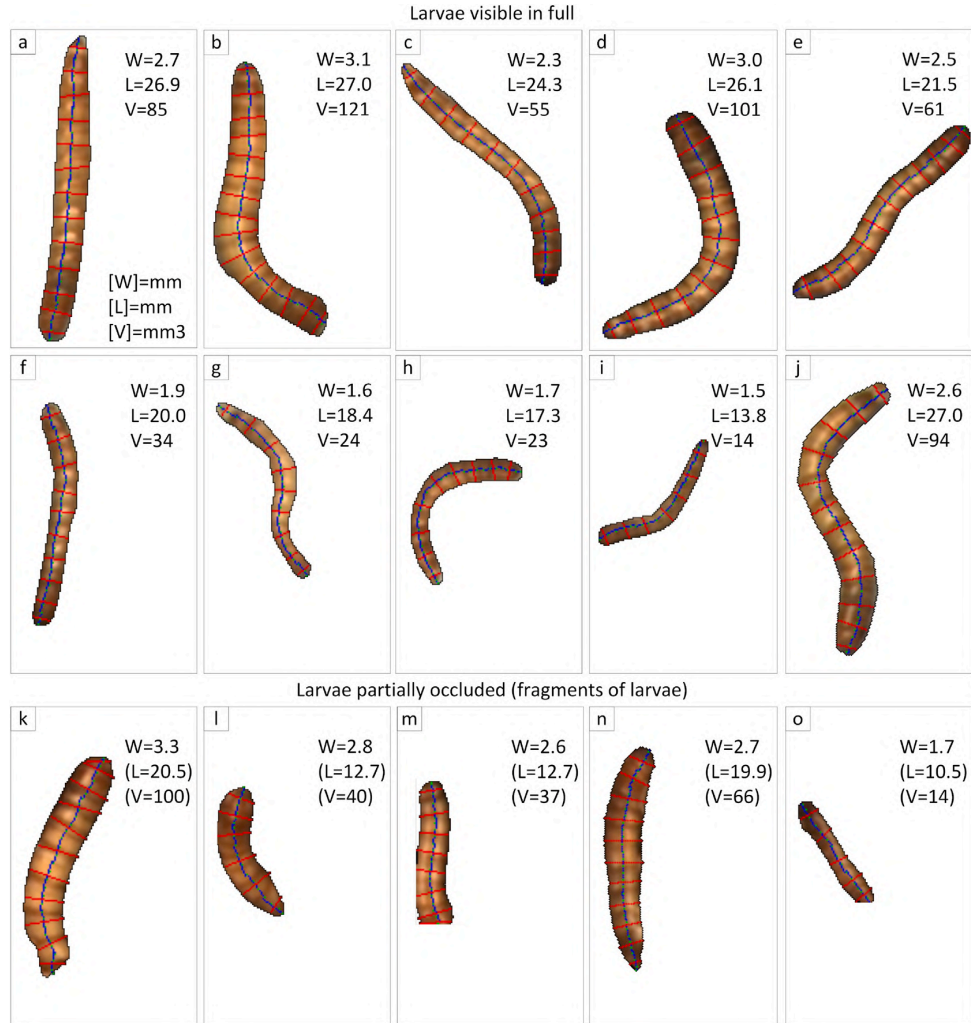


Fig. 7. Phenotyping results for individual larvae using improved multistage larvae phenotyping method for chosen objects extracted from images from dataset D1.

sections perpendicular to the skeleton can be seen at the ends of the larvae. It should be emphasised here that the definition of larval width as the median of the lengths of all red sections reduced the influence of shorter end sections on the final determined larval width. The use of width as the main measured parameter (instead of length and volume) enabled us to base only on segmented larval fragments without needing a complete mask. An interesting direction for further research is a modal segmentation models that estimate the predicted mask for invisible fragments.

### 3.2. Linear regression models length (width) and volume (width)

Characterised larvae extracted from images from the D1 dataset were used to determine length(width) and volume(width) linear regression models, presented in Fig. 8. Both relationships were characterised by a high coefficient of determination  $R^2 > 0.9$ , confirming the rationality of determining length and volume indirectly. At larger width values

(for width  $> 2.7$ ), a greater spread of values of the dependent variable (length or volume) can be observed.

### 3.3. Evaluation of larvae segmentation models for different size sets of larvae

The results of the evaluation of the larvae segmentation model for multistage larvae phenotyping are presented in Table 3. In Table 3 we can see that successive improvement steps of the larvae segmentation model enabled an increase in AP50 averaged from 75.0 to 79.2 ( $\Delta AP50 = 4.2$ ). The greatest improvement was observed for the subset of samples containing images of larvae with the shortest length (18–23 mm) - the AP50 increased from 61.7 to 72.1 ( $\Delta AP50 = 10.4$ ). For the subset of samples with images of the largest larvae (27–35 mm), the model was already characterised by a high AP50 (AP50 = 86.2) after the first step, and no significant improvement in the accuracy of



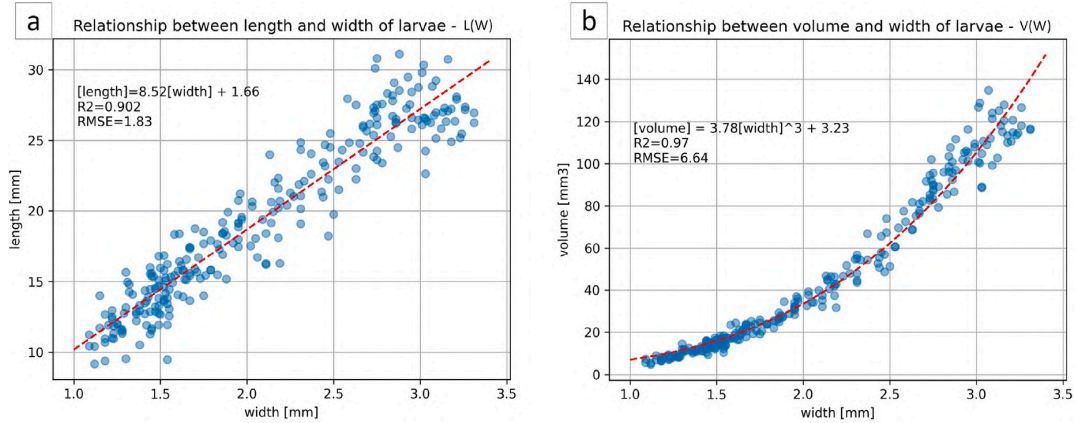


Fig. 8. Charts of the relationship (a) between length and width of larvae, and (b) between volume and width of larvae.

Table 3

Evaluation results of the larvae segmentation model for multistage phenotyping of larvae after successive development steps: (a) training the model on synthetic images (based on images from the D1 dataset), (b) training the model on synthetic images (based on images from the D3 dataset), (c) training the model on a mixed dataset (real and synthetic images based on the D3 dataset).

Approach type	Median of larvae length interval	AP50	F1-score
1. stage (only synthetic)	18–23 mm	61.7 ± 1.0	0.700 ± 0.004
1. stage (only synthetic)	23–27 mm	77.0 ± 1.1	0.793 ± 0.007
1. stage (only synthetic)	27–35 mm	86.2 ± 1.0	0.841 ± 0.006
1. stage (only synthetic)	average	75.0 ± 1.0	0.778 ± 0.006
2. stage (only synthetic)	18–23 mm	65.3 ± 2.3	0.732 ± 0.009
2. stage (only synthetic)	23–27 mm	77.0 ± 0.8	0.812 ± 0.010
2. stage (only synthetic)	27–35 mm	85.2 ± 1.0	0.843 ± 0.004
2. stage (only synthetic)	average	75.8 ± 1.4	0.796 ± 0.008
3. stage (real + synthetic)	18–23 mm	72.1 ± 1.2	0.780 ± 0.005
3. stage (real + synthetic)	23–27 mm	79.4 ± 0.6	0.834 ± 0.006
3. stage (real + synthetic)	27–35 mm	86.0 ± 0.8	0.859 ± 0.005
3. stage (real + synthetic)	average	79.2 ± 0.8	0.824 ± 0.005

this model was observed in further stages of improvement. In the context of using the developed larvae segmentation model for multistage phenotyping, it is important to note the significant difference in AP50 for the subset (18–23 mm) - AP50 = 72.1 and the subset (27–35 mm) - AP50 = 86.0. Based on these results, it can be concluded that the larvae segmentation model performed better in detecting larger larvae, which necessitated an appropriate correction when calculating quartiles for larval width. The achieved values of AP50 > 72 and F1-score > 0.77 for the larvae segmentation models were sufficient to carry out the described multistage phenotyping. The non-detection of selected larvae in dense scenes was of little relevance in the context of the problem addressed, as the number of detected instances was sufficient for the estimation of size parameters for the larvae population.

### 3.4. Dependence of the correction factor on larval width and justification of the validity of the correction

The determined calibration curve for calculating weighted quartiles is shown in Fig. 9c. In Fig. 9a and Fig. 9b, the effect of applying the correction on reducing the difference between the true larval width quartile values and those obtained after multistage phenotyping is also shown. According to preliminary assumptions, the quartile values determined during multistage phenotyping without the application of correction tended to be larger than the true quartile values (see Fig. 9a). The introduction of correction factors as weights in calculating larval width quartiles increased the  $R^2$  coefficient of determination from 0.843 to 0.927. Reducing errors in determined pseudo target values was crucial for the final accuracy of the CNN-based regressor, as

pseudo target values were used during training. In Fig. 9c, in line with initial assumptions, we observe a decreasing relationship between the correction factor and larval width. The model had particular problems for larvae characterised by width below 2.0 mm. The relatively high value of the correction factor for larval widths below 2.0 mm was because a notable proportion of small larvae were undetected by the instance segmentation model. The identified problem also influenced the overestimation of larval width quartile values in the indicated range in Fig. 9b. For the last determined point for the calibration curve, a correction factor < 1 was obtained, which may be surprising. This means that the number of objects detected with this size was greater than the true number of these objects. The reason for this situation was FP (false positive) errors, which were often in the form of 2–3 larvae detected as one in addition to the detected background elements. When analysing the determined characteristics, it should be remembered that the determined calibration curve considered both TN errors (more numerous for smaller larvae) and FP errors.

### 3.5. Evaluation and parameter fine-tuning for a regression convolutional neural network (regcnn) (custom architecture and pre-trained architecture)

The results of the evaluation of the CNN regressors are presented in Table 4 for regressors based on ImageNet pre-trained backbones and in Table 5 for regressors based on custom proposed architectures. Table 6 also shows a comparison of inference times and throughput for the different evaluated CNN-based regressor architectures.

Based on the results of the evaluation on dataset D3.TEST.1 summarised in Table 4 and in Table 5, it can be concluded that knowledge



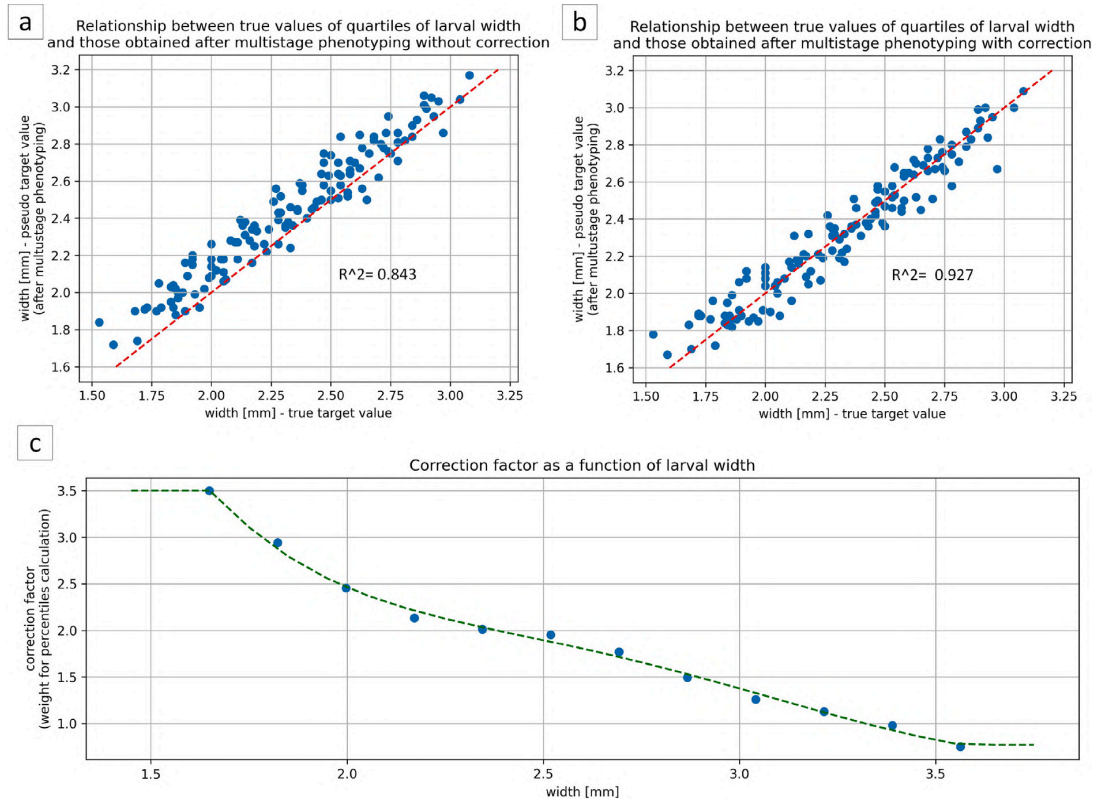


Fig. 9. Charts related to the proposed correction method: relationships between true values of quartiles of larval width and those obtained after multistage phenotyping (a) without correction, (b) with correction, and (c) correction factor as a function of larval width.

Table 4  
Evaluation results of the CNN-based regressor for pre-trained backbones on test sets D3.TEST.1 and D3.TEST.2.

Backbone name	D3.TEST.1			D3.TEST.2		
	RMSE[mm]	$R^2$	$r$	RMSE[mm]	$R^2$	$r$
ResNet18	0.090± 0.004	<b>0.934± 0.007</b>	0.967± 0.004	<b>0.131± 0.006</b>	<b>0.870± 0.013</b>	<b>0.937± 0.007</b>
ResNet50	0.093± 0.002	0.929± 0.003	0.965± 0.002	0.132± 0.002	0.867± 0.004	0.934± 0.003
ResNet101	0.125± 0.010	0.872± 0.020	0.935± 0.011	0.161± 0.011	0.802± 0.026	0.904± 0.011
MobileNetV2	0.090± 0.003	0.933± 0.005	0.967± 0.003	0.135± 0.008	0.861± 0.017	0.932± 0.009
EfficientNetB0	<b>0.089± 0.014</b>	0.934± 0.023	<b>0.968± 0.012</b>	0.135± 0.019	0.861± 0.042	0.931± 0.022
EfficientNetB4	0.101± 0.006	0.916± 0.009	0.958± 0.005	0.140± 0.005	0.850± 0.011	0.924± 0.007

Table 5  
Evaluation results of the CNN-based regressor for evaluated custom architectures on test sets D3.TEST.1 and D3.TEST.2.

Model name	D3.TEST.1			D3.TEST.2		
	RMSE[mm]	$R^2$	$r$	RMSE[mm]	$R^2$	$r$
TenebrioRegCNN_v1	0.098± 0.003	0.921± 0.004	0.961± 0.003	0.137± 0.005	0.856± 0.011	0.928± 0.005
TenebrioRegCNN_v2	0.098± 0.003	0.921± 0.005	0.960± 0.002	0.136± 0.004	0.859± 0.008	0.928± 0.006
TenebrioRegCNN_v3	<b>0.092± 0.003</b>	<b>0.930± 0.005</b>	<b>0.965± 0.002</b>	<b>0.134± 0.008</b>	<b>0.864± 0.018</b>	<b>0.930± 0.007</b>
TenebrioRegCNN_v4	0.101± 0.005	0.916± 0.008	0.959± 0.004	0.140± 0.003	0.850± 0.006	0.924± 0.003
TenebrioRegCNN_v5	0.098± 0.003	0.921± 0.005	0.960± 0.002	0.141± 0.004	0.849± 0.008	0.924± 0.005
TenebrioRegCNN_v6	0.101± 0.004	0.917± 0.007	0.958± 0.004	0.139± 0.005	0.851± 0.010	0.924± 0.005

transfer between multistage phenotyping and CNN-based regressors was reasonable. The coefficient of determination  $R^2$  on the D3.TEST.1 dataset was > 0.91 for most of the evaluated architectures (only for ResNet101 the value of this metric was lower). The evaluation results on dataset D3.TEST.2 allowed to assess the quality of the proposed end-to-end solution. Based on the evaluation results on dataset D3.TEST.2 summarised in Table 4 and in Table 5, we can observe that the best tested architecture was ResNet18, for which the following metrics were obtained: RMSE = 0.131 mm,  $R^2 = 0.870$ ,  $r = 0.937$ , which was slightly

better than for the best custom architecture (TenebrioRegCNN\_v3): RMSE = 0.134 mm,  $R^2 = 0.864$ ,  $r = 0.930$ . However, it should be noted that the TenebrioRegCNN\_v3 architecture had approximately 4.9x lower inference time per image and approximately 3.6x higher throughput (based on results in Table 6) than the ResNet18 architecture. It is reasonable to consider using custom architectures when developing low-cost solutions. The evaluation results for the ResNet18 regressor are also shown in Fig. 10, where each point in the chart represented one calculated quartile (lower quartile, median or upper

**Table 6**  
Comparison of inference time and throughput for the different evaluated CNN-based regressor architectures.

Model name	Inference time per image [ms]	Throughput [images per second]
ResNet18	11.3± 0.2	134± 1
ResNet50	35.1± 0.7	41± 1
ResNet101	56.6± 1.2	26± 1
MobileNetV2	15.1± 0.1	126± 1
EfficientNetB0	22.1± 0.1	72± 1
EfficientNetB4	55.1± 0.2	26± 1
TenebrioRegCNN_v1	1.7± 0.1	668± 3
TenebrioRegCNN_v2	4.0± 0.1	495± 3
TenebrioRegCNN_v3	2.3± 0.1	483± 1
TenebrioRegCNN_v4	5.8± 0.1	373± 1
TenebrioRegCNN_v5	1.7± 0.1	656± 1
TenebrioRegCNN_v6	3.8± 0.1	495± 1

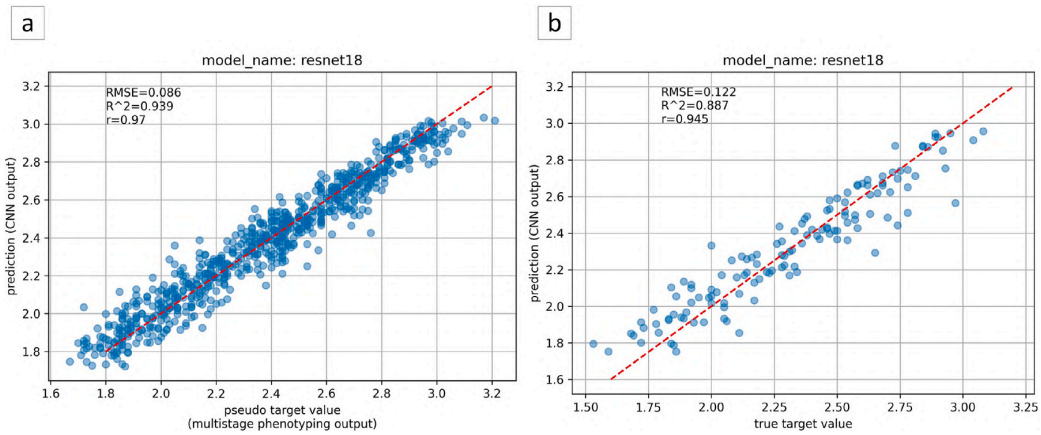


Fig. 10. Relationships between: (a) pseudo target values (output of multistage phenotyping) and CNN regressor predictions (evaluation on dataset D3.TEST.1), (b) true target values and CNN regressor predictions (evaluation on dataset D3.TEST.2) for the best obtained ResNet18 architecture, when analysing larval width.

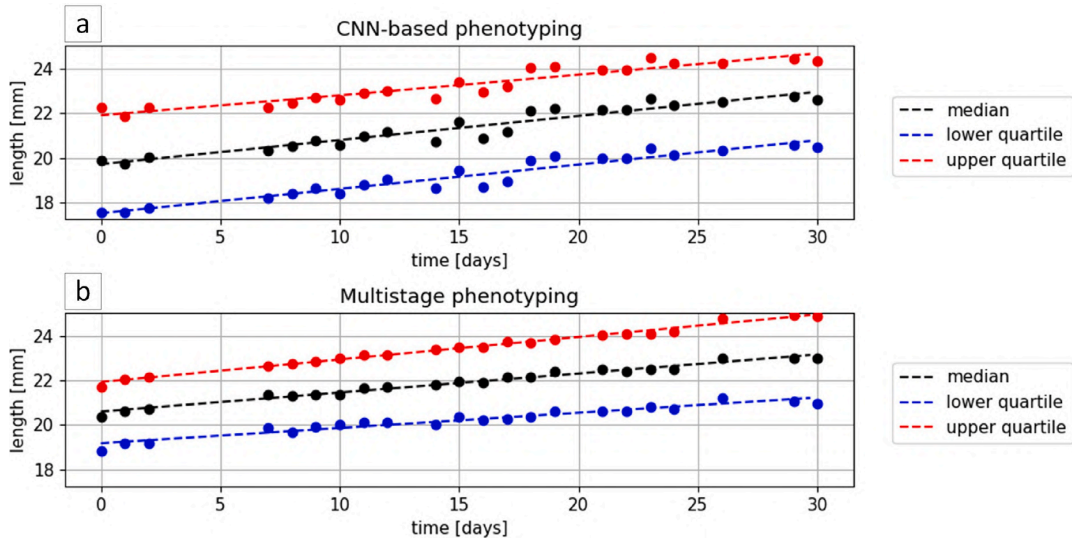


Fig. 11. Change in larval length quartiles for a chosen breeding box in a long-term breeding experiment when phenotype using: (a) regression convolutional neural network, (b) improved multistage method based on classical computer vision (standard version in Table 7).

quartile). The number of points in the charts in Fig. 10 was three times larger (due to the three quartiles determined) than the number of samples in datasets D3.TEST.1 and D3.TEST.2. The results shown in Fig. 10 refer to one particular split in the cross-validation.

By comparing the values of the coefficient of determination  $R^2$  from Fig. 9b ( $R^2 = 0.927$ ) and Fig. 10b ( $R^2 = 0.887$ ), we can estimate the level of accuracy lost in the knowledge transfer process, which should be assessed as acceptable in the context of the problem being

**Table 7**  
Comparison of the inference time of the whole pipeline in the proposed solution (CNN-based regressor) with a multistage approach and reference method.

Approach name	No. larvae for analysis	Individual larvae extraction method	Analysed area of box	$t_{mean}$	$t_{std}$	$t_{min}$	$t_{max}$
(a) CNN regressor	all	not needed (YOLOv5m for ROI assessment)	100%	<b>0.30 s</b>	0.01 s	0.28 s	0.33 s
(b) improved multistage phenotyping (standard)	all ( $\approx$ 1300 larvae)	Mask R-CNN	100%	10.93 s	1.85 s	6.02 s	13.61 s
(c) improved multistage phenotyping (limited number of analysed larvae and analysed area of box)	50	Mask R-CNN	25%	2.10 s	0.26 s	1.43 s	2.65 s
(d) reference method - multistage phenotyping (described in Majewski et al. (2022))	50	Mask R-CNN	100%	217 s	-	-	-

addressed. Despite using pseudo target values (output from multistage phenotyping) when training the CNN-based regressor, the obtained value of the coefficient of determination  $R^2 = 0.887$  was high enough that it was reasonable to replace the true target values (resulting from manual annotations) with pseudo target values when training the CNN-based regressor. Manual polygon-type annotations on images representing dense scenes are very time-consuming, and the need for labelling may occur many times when the nature of the data changes. Using pseudo target values drastically reduces the effort in developing and adapting models (re-training) to new data. The accuracy obtained, assuming the use of pseudo target values, enabled the main objective to be met, i.e. to record the size gains of the larvae in the breeding box, as confirmed by the recorded larval growth pattern in Fig. 11.

Fig. 10b confirmed the observations from the previously discussed Fig. 9b and Fig. 9c regarding the overestimation of width quartile values for larvae with widths below 2.0 mm. This limitation should be considered when analysing the developed method output.

The evaluation results on the D3.TEST.2 set, due to manual annotation, considered both the accuracy lost related to knowledge transfer and to the segmentation of the larvae. It is worth noting that the segmentation problems did not significantly reduce the performance of our solution. In addition to the dependence of segmentation accuracy on larvae size observed and extensively described in the publication, the dependence of segmentation accuracy on the density of larvae is an interesting topic for further research. In dense scenes, a single bounding box may contain many larvae, which may cause some larvae to be removed in post-processing by the Non-maximum suppression (NMS) algorithm (Neubeck and Van Gool, 2006).

### 3.6. Inference time analysis of the whole pipeline in the proposed solution

The results of inference time analysis of the whole pipeline in the proposed solution are presented in Table 7.

Based on the results in Table 7, it can be concluded that both improved multistage phenotyping and phenotyping based on a regression convolutional neural network had significantly shorter inference times than the reference method. Two versions of improved multistage phenotyping were identified. The first standard version considered the phenotyping of all larvae (about 1,300 per box) extracted from the entire box area. This approach had the best accuracy but a relatively long inference time (about 11 s/box). Within this publication, it was used to determine pseudo target values for training a CNN-based regressor. It can also be used when a small number of boxes ( $< 1000$ ) need to be analysed in laboratory breeding studies. In the second version of multistage phenotyping, the number of larvae analysed was limited to 50 and the area of the analysed boxes to 25%, adapting this method for potential use in large-scale breeding. The CNN-based regressor had the shortest inference time (0.30 s/box), favouring it for use in large-scale breeding (number of boxes  $> 10,000$ ), where inference time will be particularly important. The calculated inference time included the extraction of single larvae from images using Mask R-CNN for the multistage method and the assessment of ROI relevancy using

YOLOv5m for the CNN regressor. The percentage of total inference time spent extracting individual larvae or accessing ROIs relevance for the approaches considered is as follows: 33% for CNN regressor with ROI relevance assessment with YOLOv5m model, 63% for improved multistage phenotyping (standard) with instance segmentation with Mask R-CNN, and 85% for improved multistage phenotyping (limited number of analysed larvae and analysed area of box) with instance segmentation with Mask R-CNN. It is worth noting that for multistage phenotyping, Mask R-CNN inference time was the bottleneck of the approach, and improvements should be carried out in this part in the future. One option for speeding up processing times is to replace the Mask R-CNN model with a YOLOv8 (Jocher et al., 2023) model adapted for instance segmentation (this allows obtaining binary object masks already at the tile relevance assessment stage).

In a future comprehensive solution, it would be worth considering introducing an inference case for tiles with a small number of larvae (less than the proposed parameter  $n_{min}$ ). For such a case, it seems reasonable to carry out improved multistage phenotyping, especially as we see the potential to speed up image processing time with this approach. In view of the described possibilities to improve the multistage phenotyping approach, an increase in the value of the  $n_{min}$  parameter (above used in this study  $n_{min}=10$ ) may even be considered in the future.

### 3.7. Change of larvae size parameters in an example feeding experiment

The methods developed within this publication were used to monitor larvae growth in the breeding experiment. Charts of the change in quartiles of larvae length over time for the selected breeding box are shown in Fig. 11. In Fig. 11, results for phenotyping using a regression convolutional neural network and improved multistage phenotyping (standard version in Table 7) are shown for comparison.

In Fig. 11, we can observe smaller deviations from the trend line in the case of multistage phenotyping. In the case of CNNs, despite a larger variance for the measurement points, the correct growth trend line reconstruction is possible, which is most relevant for the experiment's performance. It is important to highlight the fact that the proposed CNN-based method was a compromise between accuracy and inference time. In the case of the standard version of multistage phenotyping, its use in large-scale breeding may not be reasonable due to the relatively long inference time.

## 4. Conclusions

The study proposed an efficient method to determine size parameters (width, length, volume) of insect larvae based on a regression convolutional neural network. The long manual annotation time of the images (due to dense scenes) was significantly reduced through knowledge transfer between improved multistage phenotyping and a CNN-based regressor. In addition, during the development of the larvae segmentation model for multistage phenotyping, generated synthetic

images and a 3-step model improvement approach were proposed. Ultimately, the solution required only a few manually annotated images for calibration. The quantitative metrics obtained for the best model (RMSE = 0.131 mm,  $R^2 = 0.870$  in larval width determination) confirmed the effectiveness of the proposed CNN-based regressor and the rationale for training the model on automatically determined pseudo target values as output from the improved multistage phenotyping. The inference time of the CNN-based regressor: 0.30 s/box meets the requirements of large-scale breeding concerning the speed of analysis allowing real-time operation.

Based on the results, we conclude that the proposed method can be successfully applied in monitoring systems for large-scale breeding of insect larvae and as a support in laboratory breeding experiments.

We consider the following as the most important directions for future work: (1) the development of models specifically adapted for inference at very low larval densities, (2) further reduction in computation time for improved multistage phenotyping, allowing this approach to be included in hybrid methods for phenotyping, (3) the development of methods for segmentation of larvae obtaining similar accuracy at different larval sizes, (4) amodal segmentation of larvae, (5) the development of reference models for the growth of insect larvae under fixed feeding, (6) the development of methods for anomaly detection based on reference models of the growth of larvae, and (7) methods for maintenance and adaptation of the proposed methods under the assumption that the nature of the data can change (domain shift).

#### CRediT authorship contribution statement

**Paweł Majewski:** Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualisation, Writing – original draft. **Mariusz Mrzygłód:** Conceptualisation, Software, Writing – review & editing. **Piotr Lampa:** Conceptualisation, Software, Visualisation, Writing – review and editing. **Robert Burduk:** Supervision, Writing – review & editing. **Jacek Reiner:** Supervision, Writing – review & editing, Project administration, Funding acquisition.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Data availability

Data will be made available on request.

#### Acknowledgements

We wish to thank Paweł Górzynski and Dawid Biedrzycki from Tenebria (Lubawa, Poland) for providing a data source of boxes with *Tenebrio molitor*. The work presented in this publication was carried out within the project “Automatic mealworm breeding system with the development of feeding technology” under Sub-measure 1.1.1 of the Smart Growth Operational Program 2014–2020 co-financed from the European Regional Development Fund on the basis of a co-financing agreement concluded with the National Center for Research and Development (NCBiR, Poland); grant POIR.01.01.01-00-0903/20.

#### References

Abbas, A., Jain, S., Gour, M., Vankudothu, S., 2021. Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Comput. Electron. Agric.* 187, 106279.

Baur, A., Koch, D., Gattermig, B., Delgado, A., 2022. Noninvasive monitoring system for *tenebrio molitor* larvae based on image processing with a watershed algorithm and a neural net approach. *J. Insects Food Feed* 8 (8), 913–920.

Bergmann, P., Fauser, M., Sattlegger, D., Steger, C., 2020. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4183–4192.

Bhoj, S., Tarafdar, A., Chauhan, A., Singh, M., Gaur, G.K., 2022. Image processing strategies for pig liveweight measurement: Updates and challenges. *Comput. Electron. Agric.* 193, 106693.

Cang, Y., He, H., Qiao, Y., 2019. An intelligent pig weights estimate method based on deep learning in sow stall environments. *IEEE Access* 7, 164867–164875.

Chaitanya, K., Erdil, E., Karani, N., Konukoglu, E., 2023. Local contrastive loss with pseudo-label based self-training for semi-supervised medical image segmentation. *Med. Image Anal.* 87, 102792.

Chaurasia, A., Culurciello, E., 2017. Linknet: Exploiting encoder representations for efficient semantic segmentation. In: *2017 IEEE Visual Communications and Image Processing. VCIP, IEEE*, pp. 1–4.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1251–1258.

Dolata, P., Wróblewski, P., Mrzygłód, M., Reiner, J., 2021. Instance segmentation of root crops and simulation-based learning to estimate their physical dimensions for on-line machine vision yield monitoring. *Comput. Electron. Agric.* 190, 106451.

EFSA Panel on Nutrition, N.F., (NDA), F.A., Turck, D., Bohn, T., Castenmiller, J., De Henauw, S., Hirsch-Ernst, K.L., Maciuk, A., Mangelsdorf, I., McArdle, H.J., Naska, A., et al., 2021. Safety of frozen and dried formulations from whole yellow mealworm (*tenebrio molitor* larva) as a novel food pursuant to regulation (EU) 2015/2283. *EFSA J.* 19 (8), e06778.

Gjergji, M., de Moraes Weber, V., Silva, L.O.C., da Costa Gomes, R., De Araújo, T.L.A.C., Pistori, H., Alvarez, M., 2020. Deep learning techniques for beef cattle body weight prediction. In: *2020 International Joint Conference on Neural Networks. IJCNN, IEEE*, pp. 1–8.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: *Advances in Neural Information Processing Systems*. Vol. 27.

Grau, T., Vilcinskis, A., Joop, G., 2017. Sustainable farming of the mealworm *tenebrio molitor* for the production of food and feed. *Zeitschrift für Naturforschung C* 72 (9–10), 337–349.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 2961–2969.

He, H., Xu, H., Zhang, Y., Gao, K., Li, H., Ma, L., Li, J., 2022. Mask R-CNN based automated identification and extraction of oil well sites. *Int. J. Appl. Earth Obs. Geoinf.* 112, 102875.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 770–778.

Houben, D., Daoulas, G., Faucon, M.-P., Dulaurent, A.-M., 2020. Potential use of mealworm frass as a fertilizer: Impact on crop growth and soil properties. *Sci. Rep.* 10 (1), 4659.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 4700–4708.

Jocher, G., Chaurasia, A., Qiu, J., 2023. Ultralytics YOLOv8. URL <https://github.com/ultralytics/ultralytics>.

Jocher, G., Nishimura, K., Mineeva, T., Vilariño, R., 2020. yolov5. p. 9, Code repository <https://github.com/ultralytics/yolov5>.

Kononov, D.A., Saleh, A., Efremova, D.B., Domingos, J.A., Jerry, D.R., 2019. Automatic weight estimation of harvested fish from images. In: *2019 Digital Image Computing: Techniques and Applications. DICTA, IEEE*, pp. 1–7.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60 (6), 84–90.

Kröncke, N., Baur, A., Böschen, V., Demtröder, S., Benning, R., Delgado, A., 2020. Automation of insect mass rearing and processing technologies of mealworms (*tenebrio molitor*). In: *African edible insects as alternative source of food, oil, protein and bioactive components*. Springer, pp. 123–139.

Li, D., Hao, Y., Duan, Y., 2020. Noninvasive methods for biomass estimation in aquaculture with emphasis on fish: A review. *Rev. Aquacult.* 12 (3), 1390–1411.

Liu, Y., Sun, P., Wergeles, N., Shang, Y., 2021. A survey and performance evaluation of deep learning methods for small object detection. *Expert Syst. Appl.* 172, 114602.

Lu, C.-Y., Rustia, D.J.A., Lin, T.-T., 2019. Generative adversarial network based image augmentation for insect pest classification enhancement. *IFAC-PapersOnLine* 52 (30), 1–5.

Majewski, P., Zapotoczny, P., Lampa, P., Burduk, R., Reiner, J., 2022. Multipurpose monitoring system for edible insect breeding based on machine learning. *Sci. Rep.* 12 (1), 1–15.

Mnih, V., Heess, N., Graves, A., et al., 2014. Recurrent models of visual attention. In: *Advances in Neural Information Processing Systems*. Vol. 27.

Neubeck, A., Van Gool, L., 2006. Efficient non-maximum suppression. In: *18th International Conference on Pattern Recognition*. Vol. 3. ICPR'06, IEEE, pp. 850–855.

Nyalala, I., Okinda, C., Kunjje, C., Korohou, T., Nyalala, L., Chao, Q., 2021. Weight and volume estimation of poultry and products based on computer vision systems: A review. *Poul. Sci.* 100 (5), 101072.

- Padilla, R., Passos, W.L., Dias, T.L., Netto, S.L., Da Silva, E.A., 2021. A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics* 10 (3), 279.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830.
- Pezzuolo, A., Guarino, M., Sartori, L., González, L.A., Marinello, F., 2018. On-barn pig weight estimation based on body measurements by a Kinect v1 depth camera. *Comput. Electron. Agric.* 148, 29–36.
- Priyadarshi, R., Rhim, J.-W., 2020. Chitosan-based biodegradable functional films for food packaging applications. *Innovat. Food Sci. Emerg. Technol.* 62, 102346.
- Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in Neural Information Processing Systems*. Vol. 28.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 4510–4520.
- Sharma, R., Biokhaghazadeh, S., Zhao, M., 2018. Are existing knowledge transfer techniques effective for deep learning on edge devices? In: *Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing*. pp. 15–16.
- Shermeyer, J., Hossler, T., Van Etten, A., Hogan, D., Lewis, R., Kim, D., 2021. Rareplanes: Synthetic data takes flight. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. pp. 207–217.
- Shin, I., Woo, S., Pan, F., Kweon, I.S., 2020. Two-phase pseudo label densification for self-training based domain adaptation. In: *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII* 16. Springer, pp. 532–548.
- Tan, M., Le, Q., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In: *International Conference on Machine Learning*. PMLR, pp. 6105–6114.
- Tsai, R.Y., 1987. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. Robot. Autom.* 3 (4), 323–344.
- Wang, Y., Mücher, S., Wang, W., Guo, L., Kooistra, L., 2023. A review of three-dimensional computer vision used in precision livestock farming for cattle growth management. *Comput. Electron. Agric.* 206, 107687.
- Weber, V.A.M., de Lima Weber, F., da Silva Oliveira, A., Astolfi, G., Menezes, G.V., de Andrade Porto, J.V., Rezende, F.P.C., de Moraes, P.H., Matsubara, E.T., Mateus, R.G., et al., 2020. Cattle weight estimation using active contour models and regression trees bagging. *Comput. Electron. Agric.* 179, 105804.
- Wu, W., Gao, X., Fan, J., Xia, L., Luo, J., Zhou, Y., 2020. Improved mask R-CNN-based cloud masking method for remote sensing images. *Int. J. Remote Sens.* 41 (23), 8910–8933.
- Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R., 2019. Detectron2. <https://github.com/facebookresearch/detectron2>.
- Yu, X., Yu, Z., Ramalingam, S., 2018. Learning strict identity mappings in deep residual networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 4432–4440.
- Zhang, T.Y., Suen, C.Y., 1984. A fast parallel algorithm for thinning digital patterns. *Commun. ACM* 27 (3), 236–239.
- Zhang, J., Zhuang, Y., Ji, H., Teng, G., 2021. Pig weight and body size estimation using a multiple output regression convolutional neural network: A fast and fully automatic method. *Sensors* 21 (9), 3218.

## 4.4 Mixing Augmentation and Knowledge-Based Techniques in Unsupervised Domain Adaptation for Segmentation of Edible Insect States

**Authors:** Paweł Majewski, Piotr Lampa, Robert Burduk, and Jacek Reiner

**Publication status:** published

**Type of publication:** conference paper

**Journal/Conference:** Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)

**MEiN points:** 70

**Lead Author:** Yes





**Corresponding Author:** Yes

**Percentage contribution:** 70%

**CRedit:** Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualisation, Writing – original draft preparation



# Mixing Augmentation and Knowledge-Based Techniques in Unsupervised Domain Adaptation for Segmentation of Edible Insect States

Paweł Majewski<sup>1</sup><sup>a</sup>, Piotr Lampa<sup>2</sup><sup>b</sup>, Robert Burduk<sup>1</sup><sup>c</sup> and Jacek Reiner<sup>2</sup><sup>d</sup>

<sup>1</sup>Faculty of Information and Communication Technology, Wrocław University of Science and Technology, Poland

<sup>2</sup>Faculty of Mechanical Engineering, Wrocław University of Science and Technology, Poland

**Keywords:** Augmentation, Domain Adaptation, Instance Segmentation, Edible Insects, *Tenebrio Molitor*.

**Abstract:** Models for detecting edible insect states (live larvae, dead larvae, pupae) are a crucial component of large-scale edible insect monitoring systems. The problem of changing the nature of the data (domain shift) that occurs when implementing the system to new conditions results in a reduction in the effectiveness of previously developed models. Proposing methods for the unsupervised adaptation of models is necessary to reduce the adaptation time of the entire system to new breeding conditions. The study acquired images from three data sources characterized by different types of cameras and illumination and checked the inference quality of the model trained in the source domain on samples from the target domain. A hybrid approach based on mixing augmentation and knowledge-based techniques was proposed to adapt the model. The first stage of the proposed method based on object augmentation and synthetic image generation enabled an increase in average  $AP_{50}$  from 58.4 to 62.9. The second stage of the proposed method, based on knowledge-based filtering of target domain objects and synthetic image generation, enabled a further increase in average  $AP_{50}$  from 62.9 to 71.8. The strategy of mixing objects from the source domain and the target domain ( $AP_{50}=71.8$ ) when generating synthetic images proved to be much better than the strategy of using only objects from the target domain ( $AP_{50}=65.5$ ). The results show the great importance of augmentation and a priori knowledge when adapting models to a new domain.


## 1 INTRODUCTION


Edible insects are one of the most promising alternative sources of novel food. The number of large-scale edible insect farms is increasing yearly due to the possibility of obtaining a high-protein product at a relatively low-cost (Dobermann et al., 2017). Edible insect breeding is a good solution for utilizing unused areas of livestock buildings where animal diseases such as ASF (African swine fever) previously occurred (Thraustardottir et al., 2021). The need to measure breeding parameters and detect anomalies, combined with the large-scale nature of breeding, necessitates using a dedicated automated monitoring system.


There have recently been few works regarding monitoring edible insect breeding related to the *Tenebrio Molitor*. (Majewski et al., 2022) proposed a multi-purpose 3-module system, enabling the detec-


tion of edible insect growth stages and anomalies (dead larvae, pests), semantic segmentation of feed, chitin, and frass, and larvae phenotyping. The authors used synthetic images generated from a pool of objects, significantly reducing model development time. Other works were based on solutions dedicated to single issues, e.g. classification of larvae segments (Baur et al., 2022), and classification of the gender of pupae (Sumriddetchkajorn et al., 2015). Undoubtedly, the results presented in this work demonstrate the feasibility of using methods based on machine learning and computer vision to inspect edible insect breeding effectively. However, adapting the developed methods to new breeding conditions is still an open problem.

In the literature, we can find a significant number of unsupervised model adaptation methods for the problems of image classification (Madadi et al., 2020), semantic segmentation (Toldo et al., 2020), or object detection (Oza et al., 2021); however, there are fewer works in the area of instance segmentation. Among the most important domain adapta-

<sup>a</sup> <https://orcid.org/0000-0001-5076-9107>

<sup>b</sup> <https://orcid.org/0000-0001-8009-6628>

<sup>c</sup> <https://orcid.org/0000-0002-3506-6611>

<sup>d</sup> <https://orcid.org/0000-0003-1662-9762>



tion methods are discrepancy-based (Csurka et al., 2017; Saito et al., 2018), adversarial-based (including generative-based) (Yoo et al., 2016; Murez et al., 2018), reconstruction-based (including graph-based) (Cai et al., 2019) and self-supervision-based (Khodabandeh et al., 2019; Shin et al., 2020). A relatively simple and intuitive approach to domain adaptation is pseudo-label-based self-training, which involves training the model for the target domain based on samples with pseudo-labels representing a prediction of the model trained on labelled samples from the source domain. An important element in this approach is prediction filtering.

The pseudo-label-based self-training approach seems suitable for instance segmentation and even easier to apply than in object detection. Namely, having masks for objects, it is possible to extract them from images, add them to appropriate object pools and use them further to generate synthetic images. It is also easier to perform filtering at the object level, as it is possible to calculate features for a specific object.

This work proposed a two-stage hybrid method for domain adaptation based on using pseudo-labels for self-training. In 1st stage, it was proposed to expand the training set of samples through augmentation at the image and object levels to reduce the overfitting of the model on the source domain. In 2nd stage, filtering of the obtained predictions was carried out using domain knowledge. An essential contribution of this work is the study of the importance of creating the training set in the 1st and 2nd stages, especially the concept of mixing real and synthetic samples and mixing samples from the source domain (with real labels) and the target domain (with pseudo-labels). In addition, the consequences of using only synthetic data (no real labelled samples in the training set) on the model's performance in cases of inference in and out of the domain were also examined.

## 2 MATERIAL AND METHODS

### 2.1 Problem Definition

The problem addressed is detection and segmentation from images of three states of edible insects, namely (1) live larvae, (2) dead larvae, and (3) pupae. The samples are in the form of 512x512 images and come from three sources associated with different types of recording cameras and lighting, namely (1) CA, (2) LU, and (3) JA. Examples of samples from the considered sources, along with the type of objects detected, are shown in Figure 1.



Figure 1: Examples of samples from the considered sources: (a) RGB images, (b) types of detected objects.

The main objective of the research was to propose a suitable domain adaptation method to train the model on one data source (source domain) with labelled samples and make inference on another (target domain) with unlabelled samples with relatively low error. The proposed method is expected to reduce the destructive effect of domain shift on the accuracy of target domain prediction.

### 2.2 Data Sources

The samples were acquired using three image acquisition systems, differing in the cameras and lighting used. The first one (CA) was an experimental station with a EOS 50D camera (Canon, Tokyo, Japan) with a resolution of 5184 x 3456 pixels and a zoom lens. Diffuse white fluorescence lighting was used. The second (LU) was a data acquisition station purposely built for imaging insects in breeding boxes. It used a Phoenix PHX120S-CC (LUCID Vision Labs, Richmond, Canada) camera with a resolution of 4096 x 3000 pixels and a 12 mm focal length lens. Samples were illuminated with neutral white LEDs in a diffusion tunnel. The third (JA) was a machine vision system prepared for industrial implementation for *Tenebrio Molitor* breeding. A GOX-12401C-PGE (JAI, Copenhagen, Denmark) camera was used, with a resolution of 4096 x 3000 pixels and a 12 mm lens. In this case, due to size limitations, LED strips providing cold white light were used for direct illumination.

### 2.3 Dataset

A dataset was prepared for the study, containing samples from various defined sources along with marked object masks from the defined classes. A total of 15 samples from CA, 29 samples from LU and 36 sam-

ples from JA were labelled. A summary of the labelled number of objects can be found in Table 1.

Table 1: The number of objects from defined classes in the considered image sources.

source type	object type	no. of objects
CA	live larvae	656
	dead larvae	250
	pupae	124
LU	live larvae	163
	dead larvae	55
	pupae	83
JA	live larvae	1247
	dead larvae	148
	pupae	187

## 2.4 Data Exploration

For initial data exploration and qualitative evaluation of domain shift, PCA and visualization of selected components were performed. The FID (Fréchet Inception Distance) metric (Heusel et al., 2017) was also calculated as a measure of the similarity of features extracted from images belonging to different sources. Lower values of the FID metric mean higher similarity of sample distributions. FID and PCA were based on a feature vector of length 2048 extracted from the last pooling layer of the deep convolutional neural network Inceptionv3 (Szegedy et al., 2015). Masked images of objects (without surrounding background) were used for feature extraction.

## 2.5 Domain Adaptation with Mixing Augmentation and Knowledge-Based Techniques

The proposed adaptation method consists of two stages described in detail in the following sections. The first stage is based on the augmentation of source domain objects and the generation of synthetic images. The second stage considers filtering target objects based on domain knowledge and re-generating synthetic images using new target domain objects. The idea scheme of the proposed solution is shown in Figure 2.

The method for generating synthetic images involved randomly placing objects on the background image and allowing the simulation of object overlap in dense scenes. Each generated synthetic image was associated with an automatically generated label. The method of generating synthetic images is described in more detail in (Majewski et al., 2022; Toda et al., 2020).

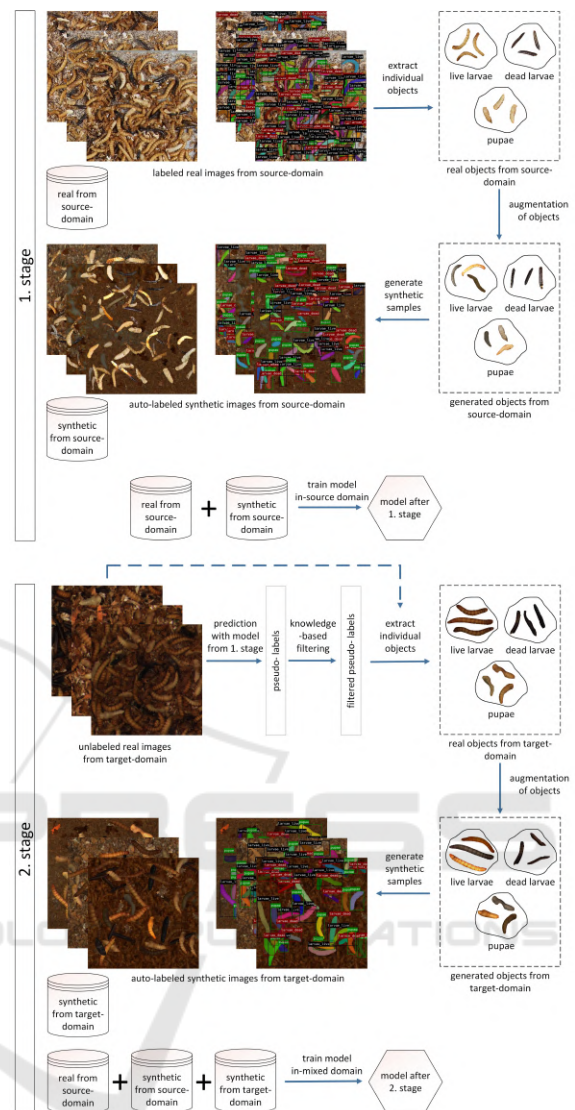


Figure 2: Idea scheme of the proposed solution detailing two stages.

### 2.5.1 First Stage of Approach with Objects Augmentation

The basis for training models is a set of real labelled samples from the source domain. Evaluation results for a model trained only on a set of real samples (the only\_real method) were used as a reference for the following proposed approaches.

Object-level augmentation and synthetic image generation were proposed to extend the source domain samples distribution. First, individual objects were extracted from real images. Then, these objects were augmented, modifying colour, contrast, sharpness and brightness. The generated objects were then placed on the background image, obtaining automati-

cally labelled synthetic images. Examples of the augmented objects and the generated synthetic images are shown in Figure 2.

Three possibilities for constructing the training set for 1st stage were identified. The `only_real` method assumes training only on real data, the `real_synthetic` method - on real and synthetic data, and the `only_synthetic` method - only on synthetic data. For each setting, the Mask-RCNN (He et al., 2017) with backbone ResNet-50 (He et al., 2016) model was trained with default training parameters. An implementation of the Mask R-CNN model from the detectron2 (Wu et al., 2019) library was used in the study. The part related to the 1st stage in Figure 2 shows the `real_synthetic` approach for creating the training set.

### 2.5.2 Second Stage of Approach with Knowledge-Based Filtering

The first component of 2nd stage of the proposed solution is an inference using the model trained in 1st stage on unlabelled target domain samples. The resulting predictions were treated as pseudo-labels that needed to be filtered to remove false positive predictions. For filtering, it was used a priori domain invariant knowledge, namely: (1) live larvae are the majority class (see in Table 1), (2) objects from the classes live larvae and dead larvae are the longest (have the largest dimension of the longer side of the bounding box), (3) objects from the dead larvae class have the lowest pixel intensity, (4) objects from the pupae class have the highest pixel intensity. The proposed knowledge-based filtering assumes successively:

1. selection of objects with a minimum length of the longer side of the bounding box  $d_{min}$ , with a predicted class live larvae,
2. removal of outliers including mean intensity, size, and length of the longer side of the bounding box among the objects extracted in 1st step, obtaining a distribution of samples representing live larvae,
3. calculation of intensity limits  $x_{min}$ ,  $x_{max}$  for samples representing live larvae,
4. selection of objects with intensity values greater than  $x_{max}$ , with predicted class pupae,
5. selection of objects with intensity values less than  $x_{min}$ , with predicted class larvae dead.
6. removal of outliers among the objects extracted in the 4th and 5th steps, obtaining a distribution of samples representing pupae and dead larvae.

The obtained new samples in the form of target domain objects and new generated objects after augmentation are then used to generate synthetic data.

In 2nd stage, we have available the following sample distributions: (1) real from the source domain, (2) synthetic from the source domain, (3) synthetic from the target domain. The study investigated the following training strategies: the "only target domain samples" strategy assumes training the model only on synthetic data from the target domain, and the "mixed source/target domain samples" strategy assumes mixing samples from the source domain and target domain in the training set. Considering the "mixed source/target domain samples" strategy, in all the approaches identified in 1st stage (`only_real`, `real_synthetic`, `only_synthetic`), the training set defined in 1st stage is extended with synthetic samples from the target domain. Figure 2 shows the "mixed source/target domain samples" strategy with the `real_synthetic` variant.

## 2.6 Evaluation

The proposed methods were evaluated using the average precision  $AP_{50}$  metric, a standard metric for the evaluation in object detection tasks. The value of the  $AP_{50}$  metric represents the area under the precision-recall curve after appropriately interpolating the chart fragments. The  $AP_{50}$  metric assumes a threshold value of intersection over union (IoU) 50% between the true and predicted bounding box to consider the prediction significant. Details regarding the calculation of the  $AP_{50}$  metric can be found in (Majewski et al., 2022; Padilla et al., 2020).

For the study, 6 possible cases of out-domain crossing (source domain  $\rightarrow$  target domain) were selected, namely CA  $\rightarrow$  LU, CA  $\rightarrow$  JA, LU  $\rightarrow$  CA, LU  $\rightarrow$  JA, JA  $\rightarrow$  CA, JA  $\rightarrow$  LU. Evaluation for the out-domain inference cases was carried out for all samples from the target domain. The  $AP_{50}$  values for in-domain inference were also determined as a reference. For the in-domain case, the entire dataset was divided into training data (80% of samples) and test data. Evaluation was performed on the test set.

## 3 RESULTS AND DISCUSSION

As part of the data exploration, PCA was performed, and FID metrics were calculated between sample distributions. A visualization of the selected components for samples from all data sources and defined classes can be found in Figure 3. The calculated FID values can be found in Table 2.

Based on the results from Figure 3 and Table 2, it can be seen that objects from the live larvae class are most similar to each other between distributions.



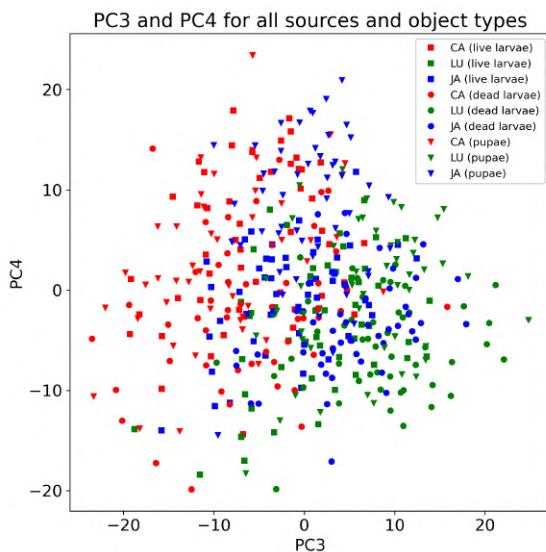


Figure 3: Selected principal components for domain shift exploration based on deep features (Inceptionv3).

Table 2: Comparison of calculated FID metrics between sources based on real samples.

sources	object type	FID
CA and LU	live larvae	124
	dead larvae	166
	pupae	144
	all (average)	145
CA and JA	live larvae	69
	dead larvae	110
	pupae	113
	all (average)	97
LU and JA	live larvae	97
	dead larvae	120
	pupae	115
	all (average)	111

The FID distances between (CA and JA) and (LU and JA) distributions are smaller than the distance between (CA and LU) distributions, which is also confirmed by Figure 3. For the selected components (PC3 and PC4), samples from JA (blue markers) are between samples from CA (red) and LU (green).

A comparison of different domain adaptation methods can be found in Table 3 for the "mixed source/target domain samples" strategy and in Table 4 for the "only target domain samples" strategy. As reference values for assessing the quality of domain adaptation are the results obtained by the models trained and tested in-domain presented in Table 5.

The results obtained for 1st stage of model adaptation (Table 3) prove that the real\_synthetic method (average  $AP_{50} = 62.9$ ), which assumes the use of both

real and synthetic samples for training, is the most suitable for use in the problem under consideration. The use of only synthetic samples (only\_synthetic method, average  $AP_{50} = 54.4$ ) or only real samples (only\_real method, average  $AP_{50} = 58.4$ ) may be better in special cases (only\_synthetic for  $LU \rightarrow CA$ , only\_real for  $LU \rightarrow JA$ ,  $CA \rightarrow JA$ ), but in general (averaged), these approaches achieve smaller AP values than the real\_synthetic method. For the special cases mentioned above, the difference between the best-obtained result and the AP value for the real\_synthetic method did not exceed  $\Delta AP_{50} = 4.0$ . On the other hand, for the  $LU \rightarrow CA$ , the difference between the AP values for only\_real and real\_synthetic was  $\Delta AP_{50} = 14.0$ , and for the  $JA \rightarrow CA$  was  $\Delta AP_{50} = 9.4$ , which is a significant difference in the effectiveness of the models.

Using only synthetic data for model training can significantly speed up the process of developing models (Majewski et al., 2022); however, based on the results obtained in this research, we can observe that this is associated with the risk of losing model accuracy. This observation is confirmed by the results after the 1st and 2nd stages of domain adaptation for inference out-domain in Table 3 (the only\_synthetic approach was characterized by  $\Delta AP_{50} = 8.5$  lower  $AP_{50}$  than the real\_synthetic approach in the 1st stage and by  $\Delta AP_{50} = 4.4$  lower  $AP_{50}$  in the 2nd second). The lack of real data in the training set mostly affects the results for in-domain inference in Table 5 ( $\Delta AP_{50} = 11.3$  difference between only\_synthetic and real\_synthetic approaches).

When considering the results separately for each of the defined classes, it should be noted that objects of the live larvae class are the easiest to detect (average  $AP_{50}$  after 2nd stage – 81.8) after a domain change, while objects of the pupae class – the most difficult (average  $AP_{50}$  after 2nd stage – 66.6). This is consistent with initial conclusions from data exploration based on FID values in Table 2.

Quantitative indicators confirm the importance of augmentation in 1st stage for the real\_synthetic approach. Additional samples complement the relevant places in the feature space and can expand the distribution of features for a given class.

Analyzing the results from 2nd stage for the two proposed strategies in Table 3 for "mixed source/target samples" strategy and in Table 4 for "only target samples" strategy, we can conclude that the "mixed source/target samples" strategy is the most suitable for creating a training set, which is confirmed by obtaining an increased  $AP_{50}$  from 65.5 to 71.8 compared to the "only target samples" strategy.

A summary of the most important results ob-

Table 3: Comparison of proposed domain adaptation methods for mixed source/target domain samples strategy.

case type	method	$AP_{50}$							
		stage 1.				stage 2. (mixing strategy)			
		live l.	dead l.	pup.	avg.	live l.	dead l.	pup.	avg.
CA → LU	only_real	64.9	58.9	74.7	66.2	79.9	59.7	75.4	71.7
	real_synthetic	70.7	61.0	78.0	<b>69.9</b>	82.8	61.7	78.1	<b>74.2</b>
	only_synthetic	63.3	54.0	65.5	60.9	82.1	62.0	77.4	73.8
CA → JA	only_real	69.7	50.4	29.2	<b>49.8</b>	75.3	55.1	36.7	55.7
	real_synthetic	72.2	38.5	27.6	46.1	76.0	55.0	36.5	<b>55.8</b>
	only_synthetic	59.3	28.6	18.6	35.5	77.3	40.8	31.7	49.9
LU → CA	only_real	41.3	58.5	39.5	46.4	79.7	78.1	69.4	<b>75.7</b>
	real_synthetic	65.1	68.8	47.2	60.4	80.2	76.4	69.6	75.4
	only_synthetic	64.3	69.3	49.9	<b>61.2</b>	79.8	76.7	70.6	<b>75.7</b>
LU → JA	only_real	73.8	53.8	28.7	<b>52.1</b>	83.3	66.9	56.2	68.8
	real_synthetic	74.1	45.9	26.2	48.7	84.9	62.3	62.9	<b>70.0</b>
	only_synthetic	59.3	31.6	14.7	35.2	83.2	50.5	55.2	63.0
JA → CA	only_real	76.8	67.1	47.8	63.9	84.6	76.4	66.8	75.9
	real_synthetic	78.8	73.3	67.7	<b>73.3</b>	82.9	76.3	69.5	<b>76.2</b>
	only_synthetic	71.4	73.6	61.1	68.7	78.5	72.8	59.8	70.4
JA → LU	only_real	75.2	68.3	71.5	71.7	83.2	74.3	84.2	<b>80.6</b>
	real_synthetic	82.9	71.5	82.7	<b>79.0</b>	84.2	70.6	82.8	79.2
	only_synthetic	74.2	60.3	60.3	64.9	79.6	61.9	73.3	71.6
all (summary)	only_real	67.0	59.5	48.6	58.4	81.0	68.4	64.8	71.4
	real_synthetic	74.0	59.8	54.9	<b>62.9</b>	81.8	67.1	66.6	<b>71.8</b>
	only_synthetic	65.3	52.9	45.0	54.4	80.1	60.8	61.3	67.4

Table 4: Results for the only target samples strategy for the 2nd stage of domain adaptation.

case type	method	$AP_{50}$			
		stage 2. (only target samples strategy)			
		live larvae	dead larvae	pupae	average
all (summary)	only_real	76.6	60.9	55.8	64.5
	real_synthetic	78.5	59.1	59.0	<b>65.5</b>
	only_synthetic	77.5	54.7	54.2	62.2

 Table 5: Reference values for domain adaptation as  $AP_{50}$  values for in-domain inference.

source type	method	$AP_{50}$			
		live larvae	dead larvae	pupae	average
all (summary)	only_real	86.6	78.8	84.4	83.3
	real_synthetic	88.4	81.8	85.2	<b>85.2</b>
	only_synthetic	80.0	71.6	70.3	73.9

tained in the study is presented on the radar plot in Figure 4. In Figure 4 it can be seen that for the crossings JA → CA, JA → LU, CA → LU, already 1st stage of the proposed method based on augmentation achieves reasonable AP results when using the real\_synthetic approach. The 2nd stage caused a significant increase in AP for the crossings LU → JA and LU → CA. After the two stages of the proposed solution, the final value of the obtained AP values strongly depended on the target domain. For crossings where the target domain was JA, the final AP values were the lowest ( $AP_{50} = 55.8$ ,  $AP_{50} = 70$ ). In summary, the

best variation of the proposed method made it possible to increase the average  $AP_{50}$  from 58.4 to 62.9 after 1st stage and to 71.8 after the 2nd stage. To obtain as high  $AP_{50}$  values as in-domain trained models ( $AP_{50} = 85.2$ ), additional labelling should be performed, especially of objects undetected by models after the 2nd stage of adaptation. The obtained  $AP_{50}$  level between 70 and 80.6 for 5 out of 6 (except for CA → JA) types of crossings between domains makes it possible to improve additional labelling by label proposals that are predictions of the model obtained after the 2nd stage.

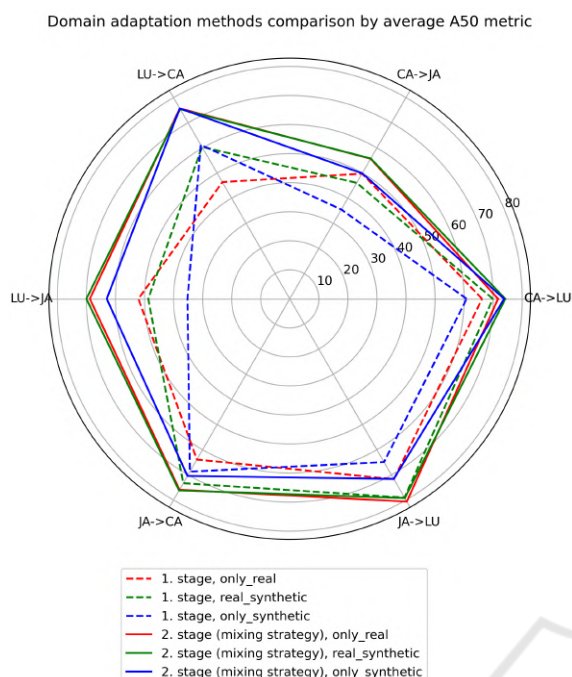


Figure 4: Comparison of proposed domain adaptation methods for different cases.

To confirm the good quality of predictions after domain adaptation, Figure 5 compares the predictions by the in-domain trained model with the predictions of the model after domain adaptation for three selected samples from different target domains. For clarity, Figure 5 shows the predictions only for the dead larvae and pupae classes.

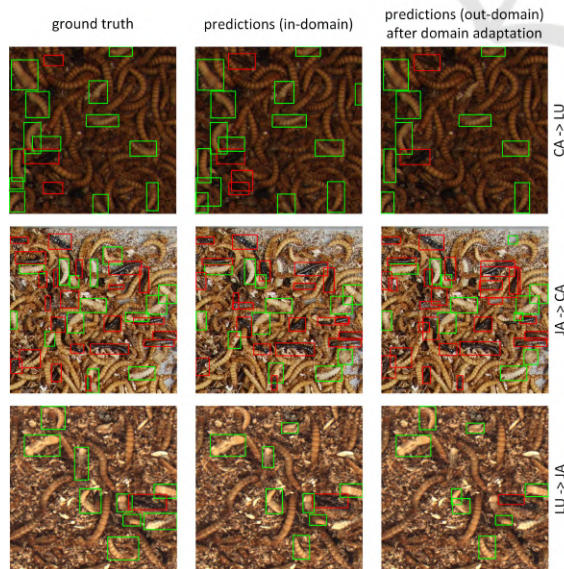


Figure 5: Comparison of predictions with ground truth for in-domain and out-domain inference cases.

## 4 CONCLUSIONS

The proposed two-stage method for domain adaptation made it possible to significantly increase the efficiency of object detection ( $AP_{50}$  increased from 58.4 to 71.8) when changing the domain without additional user supervision. The best results were obtained when the final training set consisted of real samples from the source domain, synthetic samples from the source domain and synthetic samples from the target domain (associated with filtered objects from the target domain). It confirms the validity of mixing real and synthetic samples in the training set and mixing objects from the source and target domains. It can also be concluded from the results that using only synthetic data when training models can significantly reduce the efficiency of the models for both in-domain and out-domain inference. The study showed the importance of augmentation techniques and consideration of a priori knowledge for domain adaptation.

The proposed method is flexible and can be extended to other classes of objects representing states of edible insects, e.g., beetles. The method's extension would include adding new rules when filtering the prediction based on a priori knowledge. The developed solutions will undoubtedly help rapidly adapt monitoring systems for breeding the *Tenebrio Molitor* to new breeding conditions.

Future research should focus on increasing the quality of synthetic data. An interesting research direction is to develop synthetic images based only on a priori knowledge independently of a specific domain. This approach could obtain a basic model not overfitted on a particular domain.

## ACKNOWLEDGEMENTS

We wish to thank Mariusz Mrzygłód for developing applications for the designed data acquisition workstation. We wish to thank Paweł Górczyński and Dawid Biedrzycki from Tenebria (Lubawa, Poland) for providing a data source of boxes with *Tenebrio Molitor*. The work presented in this publication was carried out within the project “Automatic mealworm breeding system with the development of feeding technology” under Sub-measure 1.1.1 of the Smart Growth Operational Program 2014-2020 co-financed from the European Regional Development Fund on the basis of a co-financing agreement concluded with the National Center for Research and Development (NCBiR, Poland); grant POIR.01.01.01-00-0903/20.



## REFERENCES

- Baur, A., Koch, D., Gatternig, B., and Delgado, A. (2022). Noninvasive monitoring system for tenebrio molitor larvae based on image processing with a watershed algorithm and a neural net approach. *Journal of Insects as Food and Feed*, pages 1–8.
- Cai, Q., Pan, Y., Ngo, C.-W., Tian, X., Duan, L., and Yao, T. (2019). Exploring object relation in mean teacher for cross-domain detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11457–11466.
- Csurka, G., Baradel, F., Chidlovskii, B., and Clinchant, S. (2017). Discrepancy-based networks for unsupervised domain adaptation: a comparative study. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2630–2636.
- Dobermann, D., Swift, J., and Field, L. (2017). Opportunities and hurdles of edible insects for food and feed. *Nutrition Bulletin*, 42(4):293–308.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.
- Khodabandeh, M., Vahdat, A., Ranjbar, M., and Macready, W. G. (2019). A robust learning approach to domain adaptive object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 480–490.
- Madadi, Y., Seydi, V., Nasrollahi, K., Hosseini, R., and Moeslund, T. B. (2020). Deep visual unsupervised domain adaptation for classification tasks: a survey. *IET Image Processing*, 14(14):3283–3299.
- Majewski, P., Zapotoczny, P., Lampa, P., Burduk, R., and Reiner, J. (2022). Multipurpose monitoring system for edible insect breeding based on machine learning. *Scientific Reports*, 12(1):1–15.
- Murez, Z., Kolouri, S., Kriegman, D., Ramamoorthi, R., and Kim, K. (2018). Image to image translation for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4500–4509.
- Oza, P., Sindagi, V. A., VS, V., and Patel, V. M. (2021). Unsupervised domain adaptation of object detectors: A survey. *arXiv preprint arXiv:2105.13502*.
- Padilla, R., Netto, S. L., and Da Silva, E. A. (2020). A survey on performance metrics for object-detection algorithms. In *2020 international conference on systems, signals and image processing (IWSSIP)*, pages 237–242. IEEE.
- Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. (2018). Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732.
- Shin, I., Woo, S., Pan, F., and Kweon, I. S. (2020). Two-phase pseudo label densification for self-training based domain adaptation. In *European conference on computer vision*, pages 532–548. Springer.
- Sumriddetchkajorn, S., Kamtongdee, C., and Chanhorm, S. (2015). Fault-tolerant optical-penetration-based silkworm gender identification. *Computers and Electronics in Agriculture*, 119:201–208.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.
- Thrastardottir, R., Olafsdottir, H. T., and Thorarinsdottir, R. I. (2021). Yellow mealworm and black soldier fly larvae for feed and food production in europe, with emphasis on iceland. *Foods*, 10(11):2744.
- Toda, Y., Okura, F., Ito, J., Okada, S., Kinoshita, T., Tsuji, H., and Saisho, D. (2020). Training instance segmentation neural network with synthetic datasets for crop seed phenotyping. *Communications biology*, 3(1):1–12.
- Toldo, M., Maracani, A., Michieli, U., and Zanuttigh, P. (2020). Unsupervised domain adaptation in semantic segmentation: a review. *Technologies*, 8(2):35.
- Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. (2019). Detectron2. <https://github.com/facebookresearch/detectron2>.
- Yoo, D., Kim, N., Park, S., Paek, A. S., and Kweon, I. S. (2016). Pixel-level domain transfer. In *European conference on computer vision*, pages 517–532. Springer.

## **4.5 Improved Pest Detection in Insect Larvae Rearing with Pseudo-Labelling and Spatio-Temporal Masking**

**Authors:** Paweł Majewski, Piotr Lampa, Robert Burduk, and Jacek Reiner

**Publication status:** published

**Type of publication:** conference paper

**Journal/Conference:** Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)

**MEiN points:** 70





**Lead Author:** Yes

**Corresponding Author:** Yes

**Percentage contribution:** 70%

**CRedit:** Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualisation, Writing – original draft preparation

# Improved Pest Detection in Insect Larvae Rearing with Pseudo-Labeling and Spatio-Temporal Masking

Paweł Majewski<sup>1</sup><sup>a</sup>, Piotr Lampa<sup>2</sup><sup>b</sup>, Robert Burduk<sup>1</sup><sup>c</sup> and Jacek Reiner<sup>2</sup><sup>d</sup>

<sup>1</sup>Faculty of Information and Communication Technology, Wrocław University of Science and Technology, Poland

<sup>2</sup>Faculty of Mechanical Engineering, Wrocław University of Science and Technology, Poland

**Keywords:** Pseudo-Labeling, Spatio-Temporal, Optical Flow, Object Detection, Insect, Monitoring, Tenebrio Molitor.

**Abstract:** Pest detection is an important application problem as it enables early reaction by the farmer in situations of unacceptable pest infestation. Developing an effective pest detection model is challenging due to the problem of creating a representative dataset, as episodes of pest occurrence under real rearing conditions are rare. Detecting the pest *Alphitobius diaperinus* Panzer in mealworm (*Tenebrio molitor*) rearing, addressed in this work, is particularly difficult due to the relatively small size of detection objects, the high similarity between detection objects and background elements, and the dense scenes. Considering the problems described, an original method for developing pest detection models was proposed. The first step was to develop a basic model by training it on a small subset of manually labelled samples. In the next step, the basic model identified low/moderate pest-infected rearing boxes from many boxes inspected daily. Pseudo-labelling was carried out for these boxes, significantly reducing labelling time, and re-training was performed. A spatio-temporal masking method based on activity maps calculated using the Gunnar-Farneback optical flow technique was also proposed to reduce the numerous false-positive errors. The quantitative results confirmed the positive effect of pseudo-labelling and spatio-temporal masking on the accuracy of pest detection and the ability to recognise episodes of unacceptable pest infestation.


## 1 INTRODUCTION


Insect pests cause significant losses in the agricultural sector every year (Oerke, 2006). Recently, an increasing consumer demand for food greenness can also be observed that favours smart solutions to control pest numbers and use chemicals, known as smart pest management (Rustia et al., 2022).


Significant advances in machine learning make researchers eager to pursue the topic of pest detection, mainly for crop pests (Li et al., 2021) and storage pests (Zhu et al., 2022). Due to the difficulty of registering pests under real-world conditions, solutions typically involved trapping pests through (1) sticky paper traps (Rustia et al., 2021), (2) pheromone-based traps (Sun et al., 2018), and (3) light traps (Bjerger et al., 2021). The machine vision system, placed at the appropriate location, enabled easy detection of


trapped pests. At the level of models/algorithms, researchers proposed different solutions, where mainly to be noted are: (1) models based on deep convolutional networks (Jiao et al., 2020; Turkoglu et al., 2022), (2) models based on transformers (Zhang et al., 2021; Wang et al., 2023) and (3) classical image processing methods (Nagar and Sharma, 2020). Among the major current challenges identified by researchers in pest detection are: (1) the difficulty of developing large datasets with issues of data augmentation and semi-supervised methods, (2) early detection of low pest infestation and indirect symptoms, (3) detection of pests when occlusion occurs, and (4) development of specific solutions, model architectures for pest detection problem as opposed to using off-the-shelf solutions (Li et al., 2021; Ngugi et al., 2021).

Despite the considerable amount of work in the area of detection of crop and storage pests, we do not find much research in the area of detection of pests in insect farming, e.g. honeybee or mealworm (*Tenebrio molitor*) (Siemianowska et al., 2013). Research has already been undertaken on detecting the mite *Varroa destructor* (Rosenkranz et al., 2010) on the bee us-

<sup>a</sup> <https://orcid.org/0000-0001-5076-9107>

<sup>b</sup> <https://orcid.org/0000-0001-8009-6628>

<sup>c</sup> <https://orcid.org/0000-0002-3506-6611>

<sup>d</sup> <https://orcid.org/0000-0003-1662-9762>

ing computer vision. (Bjerge et al., 2019) proposed an Infestation Level Estimator (ILE) to determine the level of infestation by the mite *Varroa destructor*. Despite obtaining a relatively high F1-score=0.91 for the detection of varroa mites and confirming the ability to recognise the presence of this mite on bees, the following problems of the proposed solution can be noted: (1) the significant modification of the hive to install the machine vision system, which may affect the daily functioning of the bees, (2) performing the dataset development and validation process for bee populations with relatively high infestation levels (5-10%), assuming an infestation level of 2% as an acceptable (Sajid et al., 2020). An effective pest detection solution should: (1) be designed to operate under the real conditions of farming with as little interference with insect functioning as possible, (2) be developed and evaluated for samples associated with different degrees of pest infestation in the population - the most difficult is to detect pests at low levels of infestation with an adequate level of precision (this is the situation most often found under professional farming conditions.). To the best of our knowledge, there is no work on pest detection in mealworm (*Tenebrio molitor*) rearing.

Considering the indicated research gaps at the methodological and application levels, we addressed the detection of the *Alphitobius diaperinus* Panzer pest in mealworm (*Tenebrio monitor*) rearing. To reflect the real rearing conditions fairly, the model development process used low/moderate pest-infested boxes with mealworms occurring under large-scale rearing conditions. As the main highlights of the research carried out, we identify (1) an efficient method for developing pest detection models under the assumption of low pest infestation of the population and no specially prepared samples with a high infestation, (2) a pseudo-labelling method for iteratively developing pest detection models and increasing model accuracy with relatively small manually labelled datasets, (3) a spatio-temporal masking method for increasing model precision under low pest infestation conditions, and (4) fair model evaluation under different degrees of pest infestation.

## 2 MATERIAL AND METHODS

### 2.1 Problem Definition

The problem addressed in this paper is the detection of the pest (*Alphitobius diaperinus* Panzer) in images of rearing boxes with mealworm (*Tenebrio Molitor*) larvae. The solution should include the detection of

the pest in both larva and beetle forms. The problem is challenging for the following reasons: (1) the relatively small size of the objects to be detected (the length of the mature larva is about 7 - 11 mm, and the size of the beetle is about 6 mm) (Dunford and Kaufman, 2006), (2) the high similarity between the objects to be detected and the background elements (possible false-positive errors in the case of small mealworm larvae, dead larvae), (3) dense scenes causing the objects to be detected to be often partially occluded, (4) the difficulty of developing a representative dataset containing examples of the pest under real-world conditions of mealworm rearing (breeders want to keep the pest infestation low, so the pest occurs infrequently and sparsely in rearing boxes), and (5) the labour-intensive manual labelling of images, which is directly related to the difficulties described in (1), (2) and (3). Examples of detection objects in the form of larvae (L1-L4) and beetles (B1-B3) in selected image tiles are shown in Figure 1.



Figure 1: Examples of detection objects from the classes pest larvae and pest beetle.

### 2.2 Dataset

The basis of the developed dataset was the raw 4096 x 3000 pixels images, from which were extracted smaller square tiles with size 512. The livestock-adapted machine vision system acquired raw images. The imaging conditions allowed the registration of images with a resolution of 0.143 mm/pixel. Each such image also had a corresponding image taken 1 s later, allowing further calculation of activity maps. From the raw images, 512 x 512 pixels tiles were extracted (presented in Figure 1) using the sliding window method with a shift unit of 128 pixels. For labelling, 200 rearing boxes characterised by low/moderate pest infestation levels were selected, which represented approximately 5% of all boxes being automatically inspected in a given period. A weak model (trained on a few manually labelled samples) for pest detection was used to identify boxes



with a noticeable pest infestation to avoid manual inspection. All 200 raw images were labelled manually to enable the determination of an upper baseline for the accuracy of the pest detection model, yielding the number of labelled objects: 1626 for the pest larvae class and 1004 for the pest beetle class. The average number of pests in the selected boxes, characterised by low/moderate pest infestation levels, was approximately 13. At the given level of infestation, there are more than 100 mealworm larvae per pest, which does not yet require intervention from the farmer. The dataset included 107941 tiles: 16995 tiles with at least one pest and 90946 tiles without a pest.

## 2.3 Proposed Method

Considering the difficulties described in section 2.1, an original method for developing a pest detection model is proposed. The idea scheme of the proposed solution is presented in Figure 2.

Three main elements of the proposed method are identified: (1) basic training (Figure 2a), (2) pseudo-labelling and re-training (Figure 2b), and (3) spatio-temporal masking in prediction time (Figure 2c), which will be described in the following subsections. Pseudo-labelling addressed the need to speed up (enable) the labelling of the many unlabelled images acquired during the daily inspection of the rearing boxes. Spatio-temporal masking was proposed to reduce false-positive errors, the amount of which was significant in relation to correct predictions for low/moderate pest infestations.

### 2.3.1 Basic Training

The basic training consisted of training the model on a small subset of manually labelled samples. The size of the subset was defined by the parameter *train size*, which determined approximately the proportion of all labelled objects in the training set (for example, *train size* equals 0.16 means that about 16% of all manually labelled objects representing pests were in the training set). Stratified sampling was used to maintain the proportion of objects from the pest larvae and pest beetle classes in the determined subsets of samples. The resulting model was evaluated after basic training, and the results for this type of approach were referred under the name *without pool (lower baseline)*. The name of the approach is due to the fact that unlabelled samples from the pool were not used during training. The YOLOv5x (Jocher et al., 2020) model was trained with the following training parameters: epochs=30, batch\_size=8. The basic training was presented in Figure 2a.

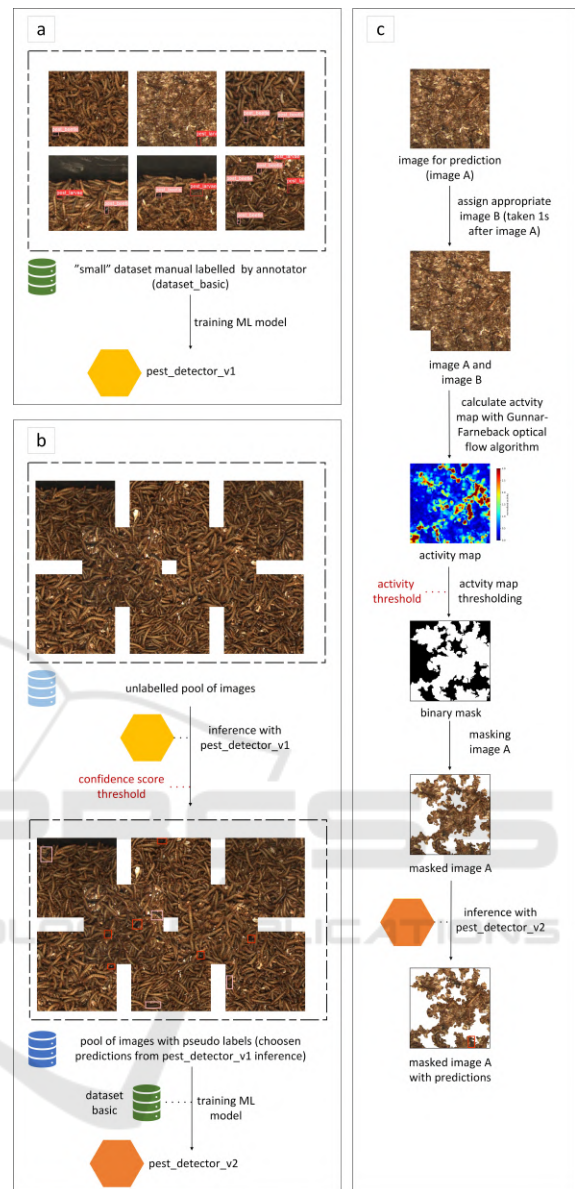


Figure 2: Idea scheme for the proposed solution: (1) basic training, (2) pseudo-labelling and re-training, and (3) spatio-temporal masking in prediction time.

### 2.3.2 Pseudo-Labelling with Re-Training

The second stage of the proposed method involved using a pseudo-labelling method to label samples from the pool automatically. The pool did not include samples selected for the test set. The inference was performed for each sample in the pool, and a prediction was considered relevant if its confidence level was higher than the parameter *confidence score threshold*. The parameter *confidence score threshold* was fine-tuned under the constant parameter *train size*. After automatic labelling according to the described

method, the model training was repeated, using the automatically labelled samples and the manually labelled samples used in the basic training. The resulting model after re-training was evaluated and the results for this approach were refereed under the name *pool used with pseudo labels*. The training settings remained unchanged. Pseudo-labelling with re-training was presented in Figure 2b.

### 2.3.3 Spatio-Temporal Masking in Prediction Time

At the prediction stage, spatio-temporal masking was introduced to remove some false-positive predictions characterised by no movement. Each image (tile) for which a prediction was performed was related to an image taken 1 s later, resulting in small shifts in the areas where the larvae were located. The normalised activity map was calculated using the Gunnar-Farneback optical flow technique (Farneback, 2003). Then, a binary mask was determined using the defined *Farneback activity threshold*, where white pixels represent areas with activity above the threshold. The *Farneback activity threshold* parameter was fine-tuned under the constant parameter *train size*. A masked RGB image was used for prediction, where only areas with the minimum defined activity are visible. When reporting the results from the model evaluations, the use of the described method was indicated by an appendix in the name + *spatio-temporal masking*. The spatio-temporal masking method was presented in Figure 2c.

## 2.4 Evaluation

Four sets of samples were distinguished for evaluation purposes: a training set, a validation set, a test set and a set defined as an image pool. Independence between the sets was provided at the level of the raw images from which the tiles were extracted. The size of the training set was defined by the parameter *train size*, which specified approximately the proportion of the number of objects in this set relative to the number of objects in the entire dataset. The training set was used to train the pest detection model. The analysis was conducted for four training set sizes: 0.02, 0.04, 0.08 and 0.16. The size of the validation set was fixed and was  $1/2$  *train size* (for example, when the training set contained about 16% of all labelled objects, the validation set then contained about 8% of all labelled objects). The validation set was used to evaluate the model during training and select the model from the best epoch. The size of the test set was fixed and equal to 0.3 (about 30% of all manually labelled objects representing pests were in the test set). The test set was

used for the final evaluation of the models, and the referenced results are from the evaluation on this set. The remaining samples not included in the training, validation and test sets belonged to the image pool. Including images from the pool in model training depended on the approach used.

Two types of evaluation were conducted for (1) low/moderate pest infestation and (2) high pest infestation. In the case of (1), the evaluation considered tiles with and without pests. For low/moderate infestation, which was present in the analysed images, there were approximately five pest-free tiles per tile with at least one pest, as described in more detail in section 2.2. In case (2), the evaluation considered only tiles with pests. It was decided to carry out these two types of evaluation because of the significant number of false-positive errors that resulted from the similarity between the analysed objects and the background elements. The possibility of numerous false-positive errors implies that the accuracy of the models will strictly depend on the level of pest infestation.

Besides evaluating the approaches indicated in section 2.3: *without pool (lower baseline)*, *pool used with pseudo labels*, an upper baseline of model accuracy was also determined by using true labels instead of pseudo labels for the pool samples. This approach was named *pool used with true labels*.

The following parameter values were checked for parameter fine-tuning procedures: (a) for the *confidence score threshold* parameter - [0.1, 0.3, 0.5, 0.7, 0.9], and for the *Farneback activity threshold* parameter - [0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2]. For the parameter *Farneback activity threshold*, the range of values was determined based on a preliminary qualitative assessment of the calculated activity maps.

For one experiment related to the selected type of evaluation, the type of approach and the size of the training set (parameter *train size*), three repeats of pest detection model training were performed related to the different division of the samples into sets: training, validation, test and image pool. The results obtained were averaged over these repeats. Repetition of training was also used in parameter fine-tuning.

Standard metrics for object detection were chosen as quantitative indicators for evaluation: AP50 (average precision with IoU=50%), F1-score, precision and recall. The values of F1-score, precision, and recall were related to the optimal working point at which the value of the F1-score metric was maximised. The values of the indicated metrics were determined separately for the two defined object classes: pest larvae and pest beetle, and averaged over these classes.



### 3 RESULTS AND DISCUSSION

A comparison of the proposed approaches for the two types of evaluation is summarised in Table 1 and in Figures 4a and 4b. In addition, Figures 3a and 3b show the results of the fine-tuning of two parameters: *confidence score threshold* and *Farneback activity threshold*. For the discussion of the results, the AP50 metric (independent of the confidence score threshold) was chosen for parameter fine-tuning and the F1-score metric (associated with a specific working point) for comparing approaches. Fine-tuning was conducted with a training set size of 0.04 and for evaluation type: low/moderate pest infestation. As lower baseline in Figure 3a the metric values for the *without pool (lower baseline)* approach were specified. In Figure 3b the lower baseline was associated with the *pool used with pseudo labels* approach. In Table 1, in addition to the value of the defined *train size* parameter, the averaged number of manually labelled samples in the training and validation set is also provided.

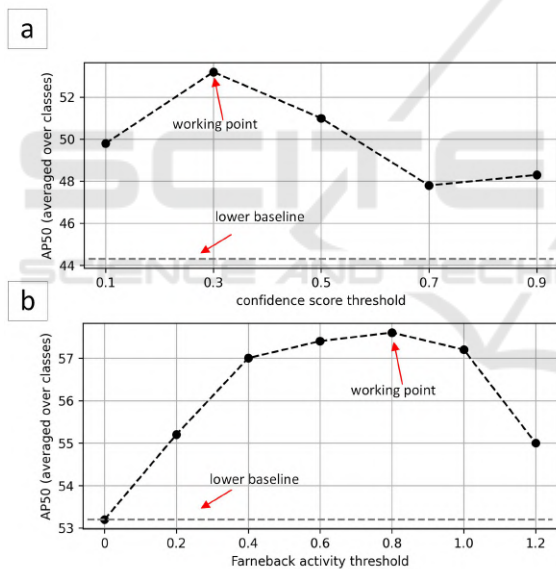


Figure 3: Fine-tuning results for: (a) confidence score threshold and (b) Farneback activity threshold.

Figure 3a and 3b confirm the rationale for fine-tuning the two selected parameters: *confidence score threshold* and *Farneback activity threshold*. For *confidence score threshold* fine-tuning, the difference between the lower baseline and the working point was  $\Delta AP50 = 8.9$  (increase from 44.3 to 53.2), while for *Farneback activity threshold*  $\Delta AP50 = 4.4$  (increase from 53.2 to 57.6). For further approaches, the parameter values indicated in Figures 3a and 3b as working points were used, i.e. 0.3 for *confidence score threshold* and 0.8 for *Farneback activity threshold*.

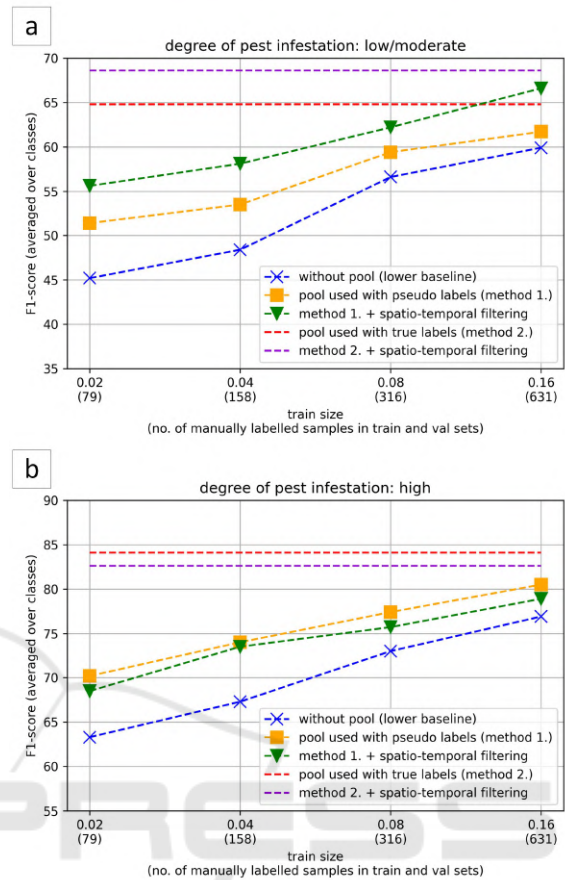


Figure 4: Comparison of the proposed methods according to the F1-score metric for pest detection for cases of: (a) low/moderate pest infestation, and (b) high pest infestation.

The impact of pseudo-labelling on pest detection accuracy can be assessed by comparing the results for approaches *without pool (lower baseline)* (blue line) and *pool used with pseudo labels* (orange line) in Figures 4a and 4b. For both low/moderate and high pest infestation, we can see a positive and significant effect of using pseudo-labelling for the image pool on pest detection accuracy. In the case of the low/moderate pest infestation evaluation, pseudo-labelling contributed to an increase in the average F1-score (averaged over different *train size*) of  $\Delta F1 = 4.0$  and in the case of the high infestation F1-score increased by  $\Delta F1 = 5.4$ .

The influence of spatio-temporal masking on detection accuracy was assessed by pairwise comparison of the results for the approaches *pool used with pseudo labels* (orange line) and *pool used with pseudo labels + spatio-temporal masking* (green line) and for the approaches *pool used with true labels* (red line) and *pool used with true labels + spatio-temporal masking* (purple line) in Figures 4a and 4b.

Table 1: Comparison of the proposed methods for two cases: (1) low/moderate pest infestation, (2) high pest infestation.

evaluation type (degree of pest infestation)	approach type	train size (# samples)	AP50	F1-score [%]	precision [%]	recall [%]
low/moderate pest infestation	without pool (lower baseline)	0.02 (79)	40.6	45.2	45.4	45.9
		0.04 (158)	44.3	48.4	50.7	47.5
		0.08 (316)	52.6	56.6	56.8	56.9
		0.16 (631)	57.1	59.9	58.9	61.2
	pool used with pseudo labels	0.02 (79)	48.8	51.4	51.5	52.1
		0.04 (158)	53.2	53.5	53.6	54.9
		0.08 (316)	57.3	59.4	57.9	61.3
		0.16 (631)	60.9	61.7	60.4	63.8
	pool used with pseudo labels + spatio-temporal filtering	0.02 (79)	51.9	55.6	57.0	55.8
		0.04 (158)	57.6	58.1	61.8	57.5
		0.08 (316)	60.9	62.2	64.7	60.9
		0.16 (631)	65.7	66.6	69.5	64.8
	pool used with true labels	all (1841)	65.3	64.8	60.7	72.0
	pool used with true labels + spatio-temporal filtering	all (1841)	68.6	68.6	68.7	69.0
high pest infestation	without pool (lower baseline)	0.02 (79)	61.5	63.3	69.6	58.9
		0.04 (158)	64.8	67.3	74.7	62.2
		0.08 (316)	72.6	73.0	79.3	68.5
		0.16 (631)	76.4	76.9	83.9	71.5
	pool used with pseudo labels	0.02 (79)	70.7	70.2	76.1	66.5
		0.04 (158)	76.2	74.0	77.3	71.7
		0.08 (316)	79.7	77.4	81.4	74.3
		0.16 (631)	83.9	80.5	82.4	79.1
	pool used with pseudo labels + spatio-temporal filtering	0.02 (79)	68.0	68.5	77.4	63.3
		0.04 (158)	74.1	73.5	81.1	68.5
		0.08 (316)	76.3	75.7	81.5	71.6
		0.16 (631)	80.5	78.9	83.6	75.4
	pool used with true labels	all (1841)	86.9	84.1	85.0	83.5
	pool used with true labels + spatio-temporal filtering	all (1841)	84.1	82.6	85.7	80.0

Considering the *pool used with pseudo labels* approach, an improvement in detection accuracy using the spatio-temporal masking technique was noted for the low/moderate pest infestation case. For this case, F1-score increased by  $\Delta F1 = 4.1$ . For the high pest infestation case, a small reduction in detection accuracy was noted - F1-score decreased by  $\Delta F1 = -1.4$ . The small reduction in detection accuracy was due to masking areas with pests characterised by low mobility. As expected, applying the spatio-temporal masking technique in general increased precision with decreasing recall. However, for the case of low/moderate pest infestation, in addition to the expected increase in precision ( $\Delta precision = 7.4$ ), an increase in recall was even observed ( $\Delta recall = 1.7$ ), which was due to the possibility of moving the working point to a lower confidence score threshold value, resulting in an increased recall. Despite the small reduction in model accuracy in the case of high pest infestation, it should be stated that this is acceptable,

considering that most boxes during the daily inspection are characterised by low/moderate pest infestation. The positive effect of spatio-temporal masking on detection accuracy is expected to be higher the smaller the pest infestation. Analogous results were obtained for the *pool used with true labels* approach, where an increase in F1-score was obtained ( $\Delta F1 = 3.8$ ) for the low/moderate infestation and a small decrease in F1-score ( $\Delta F1 = -1.5$ ) for the high pest infestation case.

Analysing the effect of training set size on detection accuracy, a significant influence of this parameter was observed in the considered range of 0.02 – 0.16. Comparing the results between *train size* 0.02 and 0.16 for *pool used with pseudo labels + spatio-temporal masking* approach (green line), an increase in F1-score was observed by  $\Delta F1 = 11.0$  for the low/moderate pest infestation case and by  $\Delta F1 = 10.4$  for the high pest infestation case. Further manual labelling of the pool samples (representing approxi-

mately 0.46 of the dataset and 1210 additional samples for manual annotation), as expected, had a positive effect on the accuracy of the models, but it was not such a spectacular improvement as in the considered range from 0.02 to 0.16. The difference between the upper baseline (the *pool used with true labels + spatio-temporal filtering* approach) and the *pool used with pseudo labels + spatio-temporal masking* approach at train size=0.16 was  $\Delta F1 = 2.0$  for the low/moderate infestation case and  $\Delta F1 = 3.7$  for the high pest infestation case, respectively. For the specific pest detection problem addressed in this article, the required minimum training set size should be at least 0.16 (associated with the validation set size 0.08), resulting in approximately 630 manually labelled objects. Assuming a low/moderate pest infestation under large-scale rearing conditions, obtaining this number of samples in a reasonable time is only possible with the support of a weak model (e.g. a model from the *pool used with pseudo labels* approach with a small train size) for identifying the boxes with the highest number of pests.

Lower metric values for the low/moderate pest infestation evaluation were obtained due to an increase in the number of false-positive predictions. Some of these predictions actually represented objects falsely detected as pests, e.g., fragments of dead larvae similar to pest beetles. A part of these false-positive predictions was filtered out by spatio-temporal masking (selected examples are shown in Figure 5).

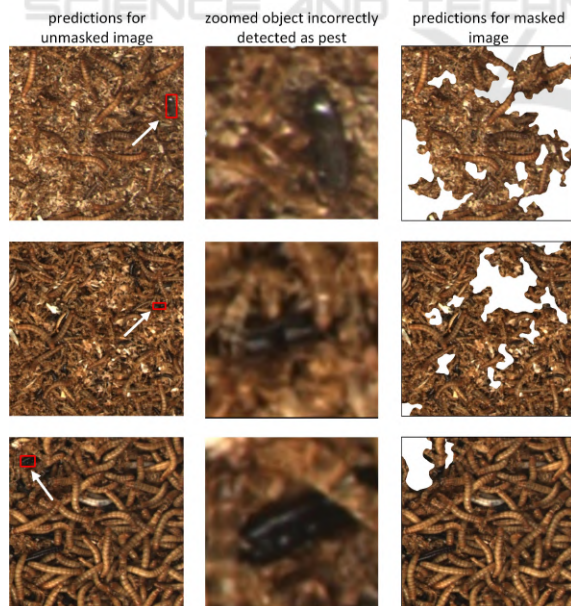


Figure 5: Examples of false-positive predictions filtered out by spatio-temporal masking.

After analysing the mistakes made by the pest detection model among the false-positive errors, we can also find many predictions that can represent not labelled pests. Some objects were difficult for the annotator to recognise, influenced by dense scenes, overlap and small size. Selected objects missed during annotation but correctly detected by the pest detection model are shown in Figure 6.

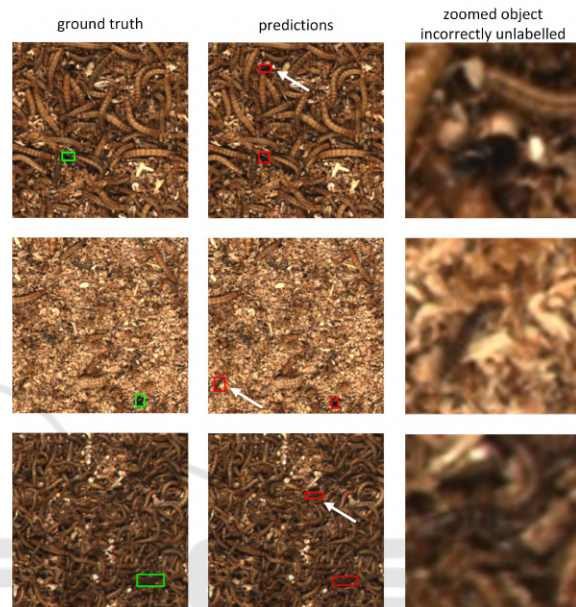


Figure 6: Selected objects missed during annotation but correctly detected by the pest detection model.

The observed problem with noisy (or lack of) labels, on the one hand, suggests that the model's accuracy can be even better than referred, and on the other hand, shows the direction of further work in label refinement.

## 4 CONCLUSIONS

The results presented here confirmed the potential of the proposed methods (pseudo-labelling, spatio-temporal masking) for developing pest detection models. Pseudo-labelling is particularly important for developing the first models (so-called weak models) when we have a small labelled dataset and access to a pool of unlabelled images. The role of the spatio-temporal masking technique is highest in the case of a low pest infestation when the main problem is potential false alarms, which is the most common situation found in professional farming. In future work, we plan to develop additional methods, e.g., based on expert knowledge and using new imaging domains to increase the precision of pest detection in the case of



low pest infestation. Future work should also analyse the real characteristics of the change in the number of pests over time when changing the infestation from low/moderate to high, which requires a fast reaction from the farmer. This analysis will enable us to improve our solution for a particular use case.

## ACKNOWLEDGEMENTS

We wish to thank Mariusz Mrzygłód for developing applications for the designed data acquisition workstation. We wish to thank Paweł Górzyński and Dawid Biedrzycki from Tenebria (Lubawa, Poland) for providing a data source of boxes with *Tenebrio Molitor*. The work presented in this publication was carried out within the project “Automatic mealworm breeding system with the development of feeding technology” under Sub-measure 1.1.1 of the Smart Growth Operational Program 2014-2020 co-financed from the European Regional Development Fund on the basis of a co-financing agreement concluded with the National Center for Research and Development (NCBiR, Poland); grant POIR.01.01.01-00-0903/20.

## REFERENCES

- Bjerge, K., Frigaard, C. E., Mikkelsen, P. H., Nielsen, T. H., Misbih, M., and Kryger, P. (2019). A computer vision system to monitor the infestation level of varroa destructor in a honeybee colony. *Computers and Electronics in Agriculture*, 164:104898.
- Bjerge, K., Nielsen, J. B., Sepstrup, M. V., Helsing-Nielsen, F., and Høye, T. T. (2021). An automated light trap to monitor moths (lepidoptera) using computer vision-based tracking and deep learning. *Sensors*, 21(2):343.
- Dunford, J. C. and Kaufman, P. E. (2006). Lesser mealworm, litter beetle, alphetobius diaperinus (panzer)(insecta: Coleoptera: Tenebrionidae): Eeny-367/in662, rev. 6/2006. *EDIS*, 2006.
- Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*, pages 363–370. Springer.
- Jiao, L., Dong, S., Zhang, S., Xie, C., and Wang, H. (2020). Af-rcnn: An anchor-free convolutional neural network for multi-categories agricultural pest detection. *Computers and Electronics in Agriculture*, 174:105522.
- Jocher, G., Nishimura, K., Mineeva, T., and Vilariño, R. (2020). yolov5. *Code repository* <https://github.com/ultralytics/yolov5>, page 9.
- Li, W., Zheng, T., Yang, Z., Li, M., Sun, C., and Yang, X. (2021). Classification and detection of insects from field images using deep learning for smart pest management: A systematic review. *Ecological Informatics*, 66:101460.
- Nagar, H. and Sharma, R. (2020). A comprehensive survey on pest detection techniques using image processing. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 43–48. IEEE.
- Ngugi, L. C., Abelwahab, M., and Abo-Zahhad, M. (2021). Recent advances in image processing techniques for automated leaf pest and disease recognition—a review. *Information processing in agriculture*, 8(1):27–51.
- Oerke, E.-C. (2006). Crop losses to pests. *The Journal of Agricultural Science*, 144(1):31–43.
- Rosenkranz, P., Aumeier, P., and Ziegelmann, B. (2010). Biology and control of varroa destructor. *Journal of invertebrate pathology*, 103:S96–S119.
- Rustia, D. J. A., Chiu, L.-Y., Lu, C.-Y., Wu, Y.-F., Chen, S.-K., Chung, J.-Y., Hsu, J.-C., and Lin, T.-T. (2022). Towards intelligent and integrated pest management through an aiot-based monitoring system. *Pest Management Science*, 78(10):4288–4302.
- Rustia, D. J. A., Lu, C.-Y., Chao, J.-J., Wu, Y.-F., Chung, J.-Y., Hsu, J.-C., and Lin, T.-T. (2021). Online semi-supervised learning applied to an automated insect pest monitoring system. *Biosystems Engineering*, 208:28–44.
- Sajid, Z. N., Aziz, M. A., Bodlah, I., Rana, R. M., Ghramh, H. A., and Khan, K. A. (2020). Efficacy assessment of soft and hard acaricides against varroa destructor mite infesting honey bee (*apis mellifera*) colonies, through sugar roll method. *Saudi Journal of Biological Sciences*, 27(1):53–59.
- Siemianowska, E., Kosewska, A., Aljewicz, M., Skibniewska, K. A., Polak-Juszczak, L., Jarocki, A., and Jedras, M. (2013). Larvae of mealworm (*tenebrio molitor* l.) as european novel food. *Agricultural Sciences*.
- Sun, Y., Liu, X., Yuan, M., Ren, L., Wang, J., and Chen, Z. (2018). Automatic in-trap pest detection using deep learning for pheromone-based dendroctonus valens monitoring. *Biosystems engineering*, 176:140–150.
- Turkoglu, M., Yanikoğlu, B., and Hanbay, D. (2022). Plant-diseasesnet: Convolutional neural network ensemble for plant disease and pest detection. *Signal, Image and Video Processing*, 16(2):301–309.
- Wang, S., Zeng, Q., Ni, W., Cheng, C., and Wang, Y. (2023). Odp-transformer: Interpretation of pest classification results using image caption generation techniques. *Computers and Electronics in Agriculture*, 209:107863.
- Zhang, Z., Gong, Z., Hong, Q., and Jiang, L. (2021). Swin-transformer based classification for rice diseases recognition. In *2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI)*, pages 153–156. IEEE.
- Zhu, L., Ma, Q., Chen, J., and Zhao, G. (2022). Current progress on innovative pest detection techniques for stored cereal grains and thereof powders. *Food Chemistry*, page 133706.

## 4.6 End-to-end Solution for *Tenebrio Molitor* Rearing Monitoring with Uncertainty Estimation and Domain Shift Detection

**Authors:** Paweł Majewski, Piotr Lampa, Robert Burduk, and Jacek Reiner

**Publication status:** accepted for publication

**Type of publication:** conference paper

**Journal/Conference:** Conference on Computer Vision and Pattern Recognition (CVPR), Agriculture-Vision Workshop: Challenges & Opportunities for Computer Vision in Agriculture

**MEiN points:** 200

**Lead Author:** Yes

**Corresponding Author:** Yes

**Percentage contribution:** 70%

**CRedit:** Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualisation, Writing – original draft preparation



# End-to-end Solution for *Tenebrio Molitor* Rearing Monitoring with Uncertainty Estimation and Domain Shift Detection

Paweł Majewski\*, Piotr Lampa, Robert Burduk, Jacek Reiner

Wrocław University of Science and Technology, Poland

\*corresponding author

pawel.majewski@pwr.edu.pl

## Abstract

*The large-scale rearing of edible insects, of which *Tenebrio Molitor* is a representative, requires monitoring using vision systems to control the process and to detect anomalies. Previously proposed solutions by researchers relied on multiple modules related to specific tasks (calculated coefficients) and specific types of models (instance segmentation, semantic segmentation). Long processing times and difficulties in maintaining and updating modules encourage the search for a more condensed solution as an end-to-end model. This paper proposed a modified YOLOv8 architecture extended with additional heads related to specific tasks. Heads were trained on problem-oriented small datasets, which significantly reduced the time spent on sample annotation. The proposed solution also included estimation of prediction uncertainty based on variation among predictions in model ensemble and detection of domain shift phenomenon. Quantitative results from the conducted experiments confirmed the potential of the developed solution.*

## 1. Introduction

Increasing demands on the quantity and quality of food produced worldwide are necessitating the search for new food sources and alternative approaches to food production [9]. Insect rearing (including edible insects) for feed and food purposes is becoming an increasingly important part of the agri-food industry. Among the most popular insect species reared for feed purposes are *Hermetia Illucens* (HI) [3] and *Tenebrio Molitor* (TM) [12]. The distinguishing factor of farming the mentioned insects is the possibility of obtaining a product rich in protein, fat and minerals at much lower environmental costs (greenhouse gas emissions, water consumption) as in the case of traditional farming (pigs, cattle) [21, 24]. The profitability of HI and TM insect farming

is closely related to its large-scale nature, which necessitates the automation of basic farming operations (e.g. feeding, harvesting) [22] on the one hand and the need for its monitoring on the other. Information obtained from data analysis is also needed to control rearing and make critical decisions, e.g. to end rearing and to change the feeding strategy.

Researchers have already addressed the problem of monitoring insect rearing on the example of the TM using a vision system and computer vision methods [16, 20]. The proposed solutions allowed (1) detection and counting of the TM growth stages (larva, pupa, beetle), (2) detection and counting of anomalies (dead larva), (3) estimation of the amount of chitinous moults and feed, and (4) estimation of size indicators of larvae (referred to as phenotyping). The developed methods were based on the following models: Mask R-CNN [13] for instance segmentation, U-Net [23] for semantic segmentation, and YOLOv5 [14] for object detection and classical image processing methods.

An undeniable disadvantage of existing solutions for TM rearing monitoring is their multi-module nature, i.e. many separate models related to a specific task. With such an approach, the problems of long image processing time (difficulty of achieving real-time inference), maintenance and updating of specific modules are of great importance. Researchers have also addressed the problem of simplifying some parts of processing, e.g., the phenotyping module, by proposing custom regression deep convolutional neural network instead of multistage image processing. However, the problem of developing a comprehensive solution should be considered still open [18].

With the above in mind, we propose an end-to-end solution for monitoring the rearing of the TM based on the YOLOv8 [15] object detection model extended with additional heads associated with a specific task (calculated indicators). To reduce labelling efforts, an approach of training individual heads on problem-oriented small datasets was proposed. Given the importance of some of the calculated

indicators in terms of rearing control and in order to increase the reliability of the solution, a method for estimating prediction uncertainties using an ensemble of models was also proposed. The calculated prediction uncertainties were also used to detect the domain shift phenomenon, which, considering the changeable conditions on the farm, is a significant problem.

## 2. Related Work

The problem of reducing multiple tasks to a single model architecture (with a shared backbone) and developing end-to-end solutions is eagerly addressed in many application areas of computer vision [2, 32], including agriculture and phenotyping of biosystems [6, 30]. Many approaches to developing condensed model architectures can be distinguished. In this section, three selected ones will be discussed, namely (1) multioutput regression models, (2) extending basic models with new heads (branches), and (3) multi-task learning.

**Multioutput regression models.** With this approach, all defined tasks are implementable by calculating a certain number of numerical values representing specific indicators. In [31], an architecture based on a backbone pre-trained on ImageNet [8] was proposed for the simultaneous calculation of six physical indicators that characterize cattle, namely the length and width of specific body parts (shoulder, hip, body) along with the estimated weight. The input to the model was recorded depth images. A combined loss based on MSE (mean squared error) was used for training, consisting of parts corresponding to prediction errors for a specific indicator. In [19], different fruit traits, i.e. moisture content (MC) and soluble solids content (SSC), were predicted simultaneously based on spectral signals from NIR spectroscopy. The proposed custom architectures consisted of a certain number of convolution layers and fully connected layers. The combined loss MSE for different coefficients was used for training as in [31].

**Extending basic models with new heads (branches).** A common approach to extend the functionality of the solution with new tasks is to extend the basic architecture with additional heads. Applying this approach, the Faster R-CNN [11] architecture was extended in [4] to include an additional branch for weight estimation. In [29], an additional block for direct counting of soybean pods was proposed as a modification to the YOLOv5 [14] model.

**Multi-task learning.** For some types of tasks, there is a need for output in the form of predictions of different types, for example, returning simultaneously bounding boxes for an object detection problem along with a predicted map for a semantic segmentation problem. For these types of issues, multi-task learning methods are helpful. The challenge in multi-task learning is to propose a suitable loss function that takes into account predictions in different formats, often with fine-tuning the weights of specific parts

in the loss function. In [5], the problem of detection and determination of cherry tomato maturity was extended to the task of detection and determination of maturity of the whole bunch. For this purpose, additional improved heads to the YOLOv7 [26] model and a combined loss function for the tasks posed were proposed. In [27], inspired by the YOLOP [28] model, a solution was proposed for the simultaneous detection of peppers, pepper segmentation and stem segmentation. The minimized loss during training consisted of three parts related to the defined tasks.

## 3. Problem Definition

The problem addressed in this paper is the calculation of multiple indicators that characterize the current status of TM rearing based on RGB images of TM rearing boxes (shown in Fig. 1).



Figure 1. Example image of a rearing box with *Tenebrio Molitor*.

The tasks undertaken include: (1) counting TM states (beetles, dead larvae and pupae), (2) estimating indicators of box coverage with chitinous moults and feed, and (3) calculating size indicators (width, length) of larvae.

Compared to the nomenclature in [16], the presented article combines object classes from the 'growth stages' and 'anomalies' groups into a single group called 'states' due to the possibility of counting objects from all classes related to TM using a single object detection model.

The counting of live larvae was abandoned from the tasks undertaken since the number of live larvae in the rearing box should be constant under normal conditions. The estimated number of live larvae will also strongly depend on the growth stage of the larvae, which is related to the influence of occlusion on the results and the tendency of larvae to hide in the substrate. With these problems present, interpreting the change in the number of live larvae over time can be problematic for the farmer.

## 4. Dataset

Multiple datasets were developed for the experiments, and each dataset was associated with a specific task. The defined datasets contained tiles of a certain size extracted from the whole image (with the size of 4096x3000 pixels) of a rearing box with *Tenebrio Molitor* as in Fig. 1.

The base dataset contained 640x640 images for training the basic YOLOv8 model to detect objects from three classes: beetles (B), dead larvae (DL) and pupae (P). Sample images with objects from the classes under consideration are shown in Fig. 2. The base dataset contained 373 images with a total number of annotations of 3442 (367 for beetles, 1781 for dead larvae and 1294 for pupae). For the base dataset, the annotations were bounding boxes.



Figure 2. Classes of detected and counted objects with example bounding boxes.

For the task of estimating the chitin coverage index (CCI) and feed coverage index (FCI), labelling was performed for 150 images with 640x640 size. Labelling consisted of marking all areas in the image representing chitin or feed. Based on the annotated images, CCI and FCI coefficients were calculated as target values. Selected samples with assigned values of CCI and FCI coefficients are shown in Fig. 3.

For the larvae phenotyping task related to calculating the three quartiles of larvae width (lower, median, upper), a dataset described in [18] consisting of 739 images of 1024x1024 size was used. Sample images from this dataset are also shown in Fig. 3.

To conduct experiments for the detection of the domain shift effect, a separate dataset was developed, consisting of images from three domains related to image registration by different vision systems (different cameras, lighting). The developed dataset contained 640x640 images, respectively 87 from the base domain, 29 from domain A and 15 from domain B. Sample images from the three considered domains are presented in Fig. 4. Details on the defined domains can also be found in [17] (data source 'JA' is the base domain, 'LU' is domain A, 'CA' is domain B).



Figure 3. Examples of samples from problem-oriented datasets for training machine learning models to proposed additional heads in YOLO architecture.

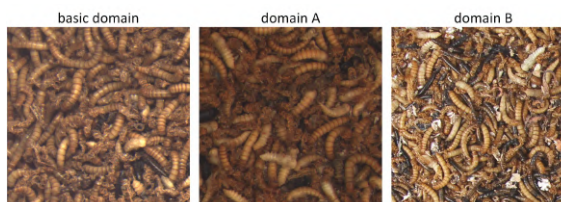


Figure 4. Examples of images from defined domains.

## 5. Proposed Approach

The proposed approach to calculating informative indicators for monitoring the rearing of TM is a modified architecture of the YOLOv8 model for object detection, which has been extended with additional problem-oriented heads, namely (1) feed coverage estimation head, (2) chitin coverage estimation head and (3) larvae phenotyping head. At the training stage, each head was separately fine-tuned using a different dataset prepared for a specific problem, saving considerable annotation time. The YOLOv8 base model allowed the detection of objects from the beetle, dead larvae and pupae classes and their counting. Feed and chitin coverage estimation heads calculated image coverage indices for feed or chitin, respectively. Coverage indices should



be understood as the number of pixels associated with the considered classes (feed or chitin) divided by the total number of pixels. In the case of the larvae phenotyping head, the output was three quartiles (lower, median, upper) of the width of the larvae as proposed in [18].

### 5.1. Model Architecture

The developed model architecture is shown in Fig. 5. The proposed three heads used features extracted from specific backbone layers of the YOLOv8 model (these are the layers with indexes from 0 to 9 shown in Fig. 5). In the case of the YOLOv8 model under consideration, the backbone is the CSPDarknet53 [25] feature extractor. The indexes of the layers used for feature extraction for the problems posed were determined experimentally, and the procedure is described in the following sections of the paper. Based on the extracted features, the selected classical machine learning models calculated the specified indices. Fig. 5 places the heads in specific locations associated with the results obtained in the conducted experiments.

### 5.2. Model Ensemble and Uncertainty Estimation

To increase the estimation accuracy of the proposed indices and enable the estimation of prediction uncertainty, an ensemble of models was considered for prediction. A bootstrap method was used to train successive models, which involved training successive YOLOv8n models on different subsets of samples determined from the basic object detection dataset. The prediction of an ensemble of models was the unweighted average of single model predictions. Uncertainty was calculated as the standard deviation among single-model predictions.

### 5.3. Domain Shift Detection

The possibility of a domain shift effect is associated with a change in the nature of the registered images. The first source of changes can be different acquisition conditions, for example, due to significant dust or contamination of the elements of the vision system. Changes can also be associated with a variation in the type of feed used or the type of rearing box used. The domain shift effect can also occur when implementing a monitoring system for a new large-scale farm.

The method for detecting the domain shift effect was based on calculated prediction uncertainties. A logistic regression model was used for the binary classification task. In addition to the standard approach of detecting domain shift for single samples, the detection of this phenomenon was also considered when averaging the uncertainty values from a subset of samples of a specific size. It was justified from the point of view of the problem addressed (registration of multiple images under large-scale rearing conditions).

## 6. Experiments

### 6.1. Selection of YOLO Core Architecture

The first stage of the conducted experiments was training YOLOv8 models using architectures with different complexity and number of parameters (n, s, m, l and x versions). The training was repeated in 5 iterations of cross-validation, where the whole dataset was divided into train/val and test parts. Model training was performed on the training set. Based on the validation set, the best training epoch was selected. On the test set, an evaluation was carried out. The best architecture was selected for further experiments based on the averaged results (metrics) obtained on the test set.

### 6.2. Selection of Best Settings for Proposed Heads

The next experiment aimed to determine the optimal settings (layer ID for feature extraction and the type of classical machine learning model for the regression task) for the proposed heads. Using the GridSearch approach, further combinations of settings were examined, whereby layers for feature extraction with indexes from 0 to 9 and the following machine learning models for regression were considered: linear regression (LR), k-nearest neighbours regression (KNN), support vector regression (SVR) [7] and gradient boosting regression (GBR) [10]. As in the first experiment, training was repeated for different iterations of cross-validation, and the results were averaged. The search for the best settings was conducted for each defined head separately. The selected best settings of each head were used for further experiments.

### 6.3. Model Ensemble and Uncertainty Estimation

The next experiment involved developing an ensemble of YOLOv8 models. For this task, the train/val and test splits from the cross-validation from the first experiment were used. Training of subsequent models was carried out on sets determined using bootstrapping. Each determined training set was extracted from the train/val part, with about 70% of the unique samples from the train/val part in the training set. To check the effect of the number of single models in the ensemble on the results, the prediction was performed in ensemble mode, averaging the single model predictions using an unweighted average. Prediction uncertainty was also determined based on the standard deviation among single model predictions in the ensemble.

### 6.4. Domain Shift Detection

The last experiment was developing a model for detecting the domain shift effect based on estimated prediction uncertainties. The Logistic Regression model was used for this task. The cases of two domains (A and B) that differed from the basic domain were considered. The obtained values of the metrics in the stratified cross-validation were referred

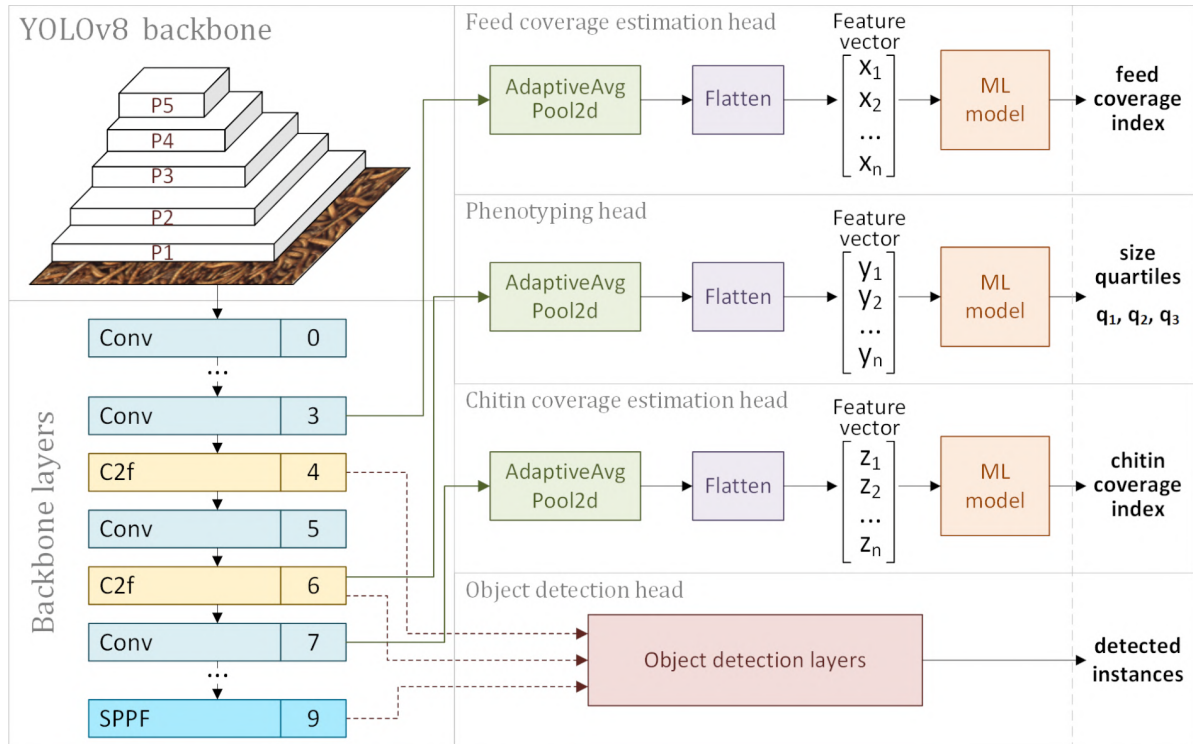


Figure 5. Modified YOLOv8 architecture with proposed additional heads: feed coverage estimation head, chitin coverage estimation head and phenotyping head.

to as the results of this experiment. The study also tested the hypothesis of the possibility of increasing the detection accuracy of the domain shift effect by averaging the prediction uncertainty over several samples. Experiments were conducted with different numbers of samples considered for averaging.

## 7. Evaluation

The proposed methods were evaluated using standard metrics for a specific problem. The referred averaged values of the specified metrics with standard deviation were based on the results obtained in successive cross-validation iterations. Consistently, the number of splits in cross-validation was set at five for all problems posed.

### 7.1. Metrics for Regression Problems

For the evaluation of regression tasks (TM states counting, estimation of chitin and feed coverage indexes), three metrics were used: mean absolute error ( $MAE$ ), coefficient of determination ( $R^2$ ) and Pearson correlation coefficient ( $r$ ), which can be calculated using formulas Eq. (1), Eq. (2), and Eq. (3).

$$MAE = \frac{1}{n_{sample}} \sum_{i=1}^{n_{sample}} |g_i - p_i| \quad (1)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n_{sample}} (g_i - p_i)^2}{\sum_{i=1}^{n_{sample}} (g_i - \bar{g})^2} \quad (2)$$

$$r = \frac{\sum_{i=1}^{n_{sample}} (g_i - \bar{g})(p_i - \bar{p})}{\sqrt{\sum_{i=1}^{n_{sample}} (g_i - \bar{g})^2} \sqrt{\sum_{i=1}^{n_{sample}} (p_i - \bar{p})^2}} \quad (3)$$

Where  $n_{sample}$  is the number of samples,  $p_i$  - prediction for the  $i$ -th sample,  $g_i$  - target value (true) for the  $i$ -th sample,  $\bar{p}$  - averaged prediction values,  $\bar{g}$  - averaged target values.

### 7.2. Metrics for Uncertainty Estimation

Evaluation of the prediction uncertainty estimation method was carried out as follows. Using a specified number of predictions in an ensemble of models, the 95 percent prediction uncertainties interval (95 PPU) was determined, calculating the lower ( $X_i^L$ ) and upper ( $X_i^U$ ) bounds of the interval being



the 2.5th and 97.5th percentiles, as in the article [1]. Having the limits of the interval, it was checked what part of the predictions fell within the determined uncertainty interval, which was referenced under the metric called *pred. in 95 PPU*. Based on the  $X_i^L$  and  $X_i^U$  values, the degree of uncertainty  $\overline{d_x}$  was also determined from the formula Eq. (4) and then the d-factor metric from the formula Eq. (5).

$$\overline{d_x} = \frac{1}{n_{sample}} \sum_{i=1}^{n_{sample}} (X_i^U - X_i^L) \quad (4)$$

$$d - factor = \frac{\overline{d_x}}{\sigma_x} \quad (5)$$

Where  $\sigma_x$  is the standard deviation among the target values for the selected problem

### 7.3. Metrics for Domain Shift Detection

To evaluate domain shift detection models, precision, recall and F1-score metrics were used, whose formulas can be found in Eq. (6), Eq. (7) and Eq. (8).

$$precision = \frac{TP}{TP + FP} \quad (6)$$

$$recall = \frac{TP}{TP + FN} \quad (7)$$

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (8)$$

Where TP, TN, FP and FN represent the number of true positive, true negative, false positive and false negative predictions, respectively.

### 7.4. Inference Time

The referenced inference time values were based on predictions made using hardware with the following specifications: GeForce RTX 2060 SUPER 8 GB (GPU) and AMD Ryzen 7 1700 3 GHz (CPU). When referencing inference times for the entire rearing box (size of 4096x3000), the inference was assumed for 54 individual tiles (dividing the entire image into 640x640 tiles with 25% overlap).

## 8. Results and Discussion

### 8.1. Selection of YOLO Core Architecture

In the first step of developing the proposed solution, the appropriate architecture of the YOLOv8 model was selected, the results of which are presented in Tab. 1.

Based on the results obtained in Tab. 1, it was decided to select the YOLOv8n architecture for further experiments. The YOLOv8n architecture had the highest metrics for the counting task and the highest throughput, which is particularly important for inference in model ensemble mode.

model	class	MAE	$R^2$	$r$
YOLOv8n	B	<b>0.20</b> $\pm 0.04$	<b>0.959</b> $\pm 0.011$	<b>0.985</b> $\pm 0.007$
YOLOv8n	DL	<b>1.07</b> $\pm 0.21$	<b>0.908</b> $\pm 0.020$	<b>0.962</b> $\pm 0.005$
YOLOv8n	P	<b>0.82</b> $\pm 0.05$	0.969 $\pm 0.012$	0.988 $\pm 0.006$
YOLOv8s	B	0.21 $\pm 0.06$	0.955 $\pm 0.009$	0.982 $\pm 0.004$
YOLOv8s	DL	1.19 $\pm 0.09$	0.893 $\pm 0.019$	0.955 $\pm 0.002$
YOLOv8s	P	0.88 $\pm 0.10$	0.966 $\pm 0.011$	0.988 $\pm 0.006$
YOLOv8m	B	0.21 $\pm 0.08$	0.949 $\pm 0.020$	0.977 $\pm 0.012$
YOLOv8m	DL	1.14 $\pm 0.12$	0.897 $\pm 0.022$	0.955 $\pm 0.009$
YOLOv8m	P	0.85 $\pm 0.14$	0.966 $\pm 0.017$	<b>0.989</b> $\pm 0.005$
YOLOv8l	B	0.21 $\pm 0.07$	0.956 $\pm 0.032$	0.979 $\pm 0.017$
YOLOv8l	DL	1.26 $\pm 0.15$	0.888 $\pm 0.022$	0.951 $\pm 0.008$
YOLOv8l	P	0.83 $\pm 0.09$	0.969 $\pm 0.015$	0.987 $\pm 0.008$
YOLOv8x	B	0.23 $\pm 0.08$	0.954 $\pm 0.012$	0.982 $\pm 0.007$
YOLOv8x	DL	1.22 $\pm 0.22$	0.889 $\pm 0.012$	0.953 $\pm 0.010$
YOLOv8x	P	0.85 $\pm 0.05$	<b>0.973</b> $\pm 0.010$	0.988 $\pm 0.004$

Table 1. Results for Tenebrio Molitor states (beetle/B, dead larva/DL, pupa/P) counting for different types of YOLO models.

It is noteworthy that already at the level of the object detection model, it was possible to achieve a significant reduction in computation time compared to [16], where the YOLOv5x model characterized by an inference time of 40 ms/tile was used. In the case of the YOLOv8n model, the inference time was 7.9 ms/tile. The reduction in computation time would be even greater assuming batch inference (the throughput for YOLOv8n was 395 tiles/s). This results in a computation time of about 0.14s for the entire rearing box (composed of 54 tiles). With such values of processing times, even ensemble mode inference, with a reasonable number of single models, is reasonable.

### 8.2. Selection of Best Settings for Proposed Heads

In the next step, the best settings (machine learning model for regression and layer ID for feature extraction) were searched for the proposed additional heads. The results from this step are presented in Tab. 2.

Based on the results in Tab. 2, it can be concluded that different models and features extracted from different layers were the best choice for different tasks. Finally, for the chitin coverage estimation head, the GBR model based on features extracted from the 7th layer was chosen; for the feed coverage estimation head - the LR model and features from the 3rd layer; and for the phenotyping head - the GBR model along with features from the 6th layer. The relatively high results ( $R^2 > 0.78$ ) confirmed the validity of the proposed solution based on attaching additional heads to the base YOLOv8n model. The lowest results were achieved for the estimation of the chitin coverage index. This may be due to the high similarity between live larvae and chitinous moults. It is noteworthy that the results obtained for the

head	model	layer	MAE	$R^2$	$r$
chitin	LR	3	0.082 $\pm$ 0.024	0.752 $\pm$ 0.128	0.919 $\pm$ 0.031
chitin	KNN	9	0.075 $\pm$ 0.023	0.716 $\pm$ 0.185	0.911 $\pm$ 0.036
chitin	SVR	4	0.070 $\pm$ 0.018	<b>0.786</b> $\pm$ 0.130	0.947 $\pm$ 0.022
chitin	GBR	7	<b>0.062</b> $\pm$ 0.025	0.785 $\pm$ 0.169	<b>0.948</b> $\pm$ 0.030
feed	LR	3	<b>0.042</b> $\pm$ 0.008	<b>0.949</b> $\pm$ 0.025	<b>0.983</b> $\pm$ 0.007
feed	KNN	6	0.065 $\pm$ 0.017	0.857 $\pm$ 0.113	0.938 $\pm$ 0.042
feed	SVR	0	0.046 $\pm$ 0.009	0.941 $\pm$ 0.026	0.976 $\pm$ 0.012
feed	GBR	6	0.065 $\pm$ 0.021	0.850 $\pm$ 0.137	0.945 $\pm$ 0.038
pheno	LR	6	0.106 $\pm$ 0.005	0.863 $\pm$ 0.012	0.930 $\pm$ 0.007
pheno	KNN	9	0.119 $\pm$ 0.008	0.829 $\pm$ 0.023	0.917 $\pm$ 0.010
pheno	SVR	6	<b>0.101</b> $\pm$ 0.002	0.868 $\pm$ 0.008	0.932 $\pm$ 0.004
pheno	GBR	6	0.103 $\pm$ 0.004	<b>0.869</b> $\pm$ 0.012	<b>0.935</b> $\pm$ 0.006

Table 2. Results for the tasks related to the proposed heads, i.e., chitin coverage estimation (chitin), feed coverage estimation (feed) and larvae phenotyping (pheno) using different settings (chosen machine learning models for prediction based on embeddings from a specific layer of the YOLO model).

phenotyping head are comparable with the results reported in [18], where a special architecture was used for the task of phenotyping larvae based on the ResNet18 model with fine-tuning of all model parameters. In the approach considered in this article, we assume frozen weights for the backbone.

### 8.3. Predictions with Proposed Heads

The evaluation results in the form of true versus predicted charts for the regression tasks of estimating feed coverage index and larvae phenotyping are shown in Fig. 6

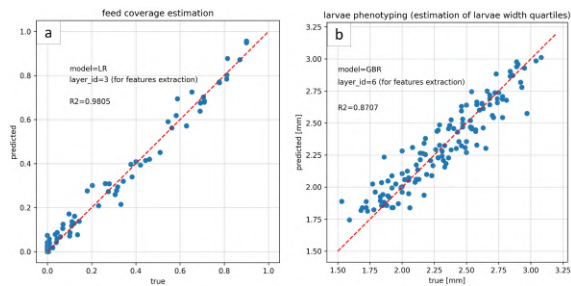


Figure 6. Comparative analysis of true vs. predicted values for selected regression tasks: (a) feed coverage estimation and (b) phenotyping based on the results from the selected cross-validation iteration.

The results in Fig. 6 confirm the validity of the developed approach to calculating the proposed indices and indicate the great potential of extracted features from the chosen backbone layers of the YOLOv8n model.

### 8.4. Model Ensemble and Uncertainty Estimation

The results for prediction in model ensemble mode for the Tenebrio Molitor state counting task are shown in Tab. 3 in the context of prediction efficiency and Tab. 4 for estimation of prediction uncertainty.

mode	class	MAE	$R^2$	$r$
single model	B	0.212 $\pm$ 0.082	0.949 $\pm$ 0.040	0.981 $\pm$ 0.011
single model	DL	1.134 $\pm$ 0.155	0.898 $\pm$ 0.021	0.955 $\pm$ 0.009
single model	P	0.906 $\pm$ 0.166	0.965 $\pm$ 0.023	0.987 $\pm$ 0.008
ensemble(n=5)	B	0.194 $\pm$ 0.055	0.967 $\pm$ 0.013	0.987 $\pm$ 0.006
ensemble(n=5)	DL	1.004 $\pm$ 0.120	0.924 $\pm$ 0.013	0.965 $\pm$ 0.006
ensemble(n=5)	P	0.759 $\pm$ 0.087	0.979 $\pm$ 0.012	0.991 $\pm$ 0.006
ensemble(n=10)	B	0.189 $\pm$ 0.054	0.970 $\pm$ 0.011	0.988 $\pm$ 0.005
ensemble(n=10)	DL	0.989 $\pm$ 0.114	0.927 $\pm$ 0.012	0.966 $\pm$ 0.006
ensemble(n=10)	P	0.732 $\pm$ 0.08	0.981 $\pm$ 0.011	0.991 $\pm$ 0.006
ensemble(n=20)	B	0.188 $\pm$ 0.054	0.971 $\pm$ 0.011	0.988 $\pm$ 0.005
ensemble(n=20)	DL	0.982 $\pm$ 0.114	0.929 $\pm$ 0.010	0.966 $\pm$ 0.005
ensemble(n=20)	P	0.718 $\pm$ 0.075	0.981 $\pm$ 0.012	0.991 $\pm$ 0.006

Table 3. Comparison of results for single-model and ensemble of models approaches for Tenebrio Molitor states counting.

mode	class	pred. in 95 PPU	d-factor
ensemble(n=5)	B	0.928 $\pm$ 0.029	0.119 $\pm$ 0.032
ensemble(n=5)	DL	0.714 $\pm$ 0.040	0.251 $\pm$ 0.045
ensemble(n=5)	P	0.777 $\pm$ 0.046	0.138 $\pm$ 0.019
ensemble(n=10)	B	0.956 $\pm$ 0.026	0.156 $\pm$ 0.034
ensemble(n=10)	DL	0.811 $\pm$ 0.030	0.328 $\pm$ 0.049
ensemble(n=10)	P	0.859 $\pm$ 0.037	0.180 $\pm$ 0.023
ensemble(n=20)	B	0.975 $\pm$ 0.017	0.185 $\pm$ 0.044
ensemble(n=20)	DL	0.865 $\pm$ 0.010	0.381 $\pm$ 0.053
ensemble(n=20)	P	0.920 $\pm$ 0.019	0.214 $\pm$ 0.023

Table 4. Results for prediction uncertainty estimation using model ensemble for Tenebrio Molitor states counting.

Based on the results in Tab. 3, it can be concluded that, as expected, using an ensemble of YOLOv8 models resulted in a significant increase in counting performance compared to the results achieved by single models. The optimal number of models for the ensemble is not obvious. On the one hand, increasing the number of models in the ensemble from 10 to 20 no longer resulted in a significant increase in prediction accuracy. On the other hand, based on the results in Tab. 4, we can see that using more models in an ensemble results in more accurate uncertainty estimation (a larger proportion of predictions bracketed by 95 PPU). Of course, this is also related to the larger d-factor associated with the size of the uncertainty interval. The final decision on the number of models for the ensemble should be made, taking into ac-

count the characteristics of the problems, that is, the cost of potential FP and FN errors.

### 8.5. Domain Shift Detection

The distributions of prediction uncertainties for samples from the defined domains are shown in Fig. 7

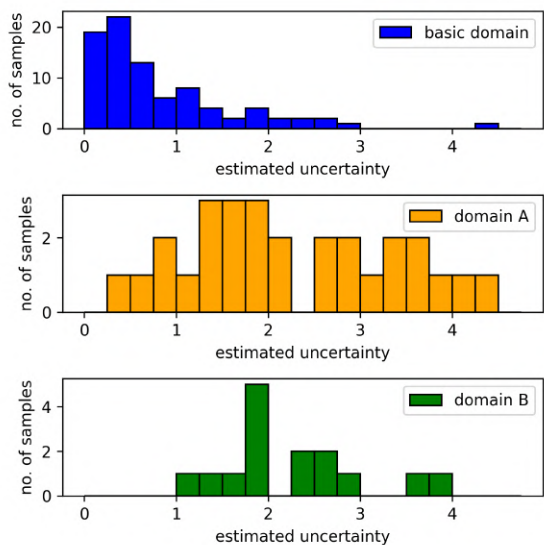


Figure 7. Comparison of the distributions of estimated uncertainty for samples from the base domain and samples from other domains (A and B).

In Fig. 7, we can see that considering the uncertainties for single samples, there is a noticeable overlap between the distributions for the defined domains. Considering the significant difference between the average uncertainty for the considered distributions, averaging the uncertainty values for a subset of samples of a certain size can significantly improve the separability of the distributions. We can find confirmation of this hypothesis in Tab. 5, where quantitative results are presented for detecting the domain shift phenomenon at a certain size of the subset of samples used to calculate the averaged uncertainty.

With 10 samples taken for averaged uncertainty,  $F1 > 0.94$  was achieved for the two domains considered. Quantitative indicators confirm the validity of detecting the phenomenon of domain shift in the proposed way based on an increase in the value of prediction uncertainty. Carrying out a procedure for detecting the phenomenon of domain shift in production conditions for the problem posed also does not seem complicated, given the thousands of images recorded daily (which may represent a subset for averaging uncertainty).

set size	new domain	F1	precision	recall
1	A	0.622 $\pm$ 0.071	0.573 $\pm$ 0.067	0.693 $\pm$ 0.118
1	B	0.620 $\pm$ 0.104	0.487 $\pm$ 0.086	0.867 $\pm$ 0.163
5	A	0.833 $\pm$ 0.140	0.787 $\pm$ 0.181	0.893 $\pm$ 0.088
5	B	0.876 $\pm$ 0.123	0.800 $\pm$ 0.187	1.000 $\pm$ 0.000
10	A	0.945 $\pm$ 0.078	0.971 $\pm$ 0.057	0.933 $\pm$ 0.133
10	B	0.971 $\pm$ 0.057	0.950 $\pm$ 0.100	1.000 $\pm$ 0.000

Table 5. Results for domain shift detection for different sizes of the subset of samples used for averaged uncertainty calculation.

## 9. Conclusion and Future Work

The research proposed an end-to-end solution for calculating indicators to support the monitoring of *Tenebrio Molitor* rearing. Compared to previous approaches, multiple (separate) models trained for specific tasks were reduced to a single architecture based on a shared backbone. The extended YOLOv8 architecture with three problem-oriented heads made it possible to perform predictions for specific regression tasks. Training for each head was done separately, which made it possible to develop smaller datasets focused on defined object classes and significantly reduce the time spent on labelling. The proposed solution is flexible and allows rapid architecture extension for new problems by adding the following heads. Using an ensemble of models made it possible to increase the accuracy of prediction and estimate the uncertainty of prediction, which will increase the reliability of the developed solution and facilitate critical decision-making on the farm.

Future work should focus on multi-task learning problems, making it possible to jointly learn the separated heads of the architecture. Given the dependencies between the calculated indicators (e.g., the occurrence of pupae is related to a certain size of larvae), this approach seems reasonable.

## Acknowledgements

We wish to thank Professor Ta-Te Lin (Department of Biomechanics Engineering, National Taiwan University, Taiwan, ROC) for his valuable comments on the developed solution and review. We wish to thank Paweł Górczyński and Dawid Biedrzycki from *Tenebria* (Lubawa, Poland) for providing a data source of boxes with *Tenebrio molitor*. The work presented in this publication was carried out within the project “Automatic mealworm breeding system with the development of feeding technology” under Sub-measure 1.1.1 of the Smart Growth Operational Program 2014–2020 co-financed from the European Regional Development Fund on the basis of a co-financing agreement concluded with the National Center for Research and Development (NCBiR, Poland); grant POIR.01.01.01-00-0903/20.

## References

- [1] Chetan Badgajar, Daniel Flippo, and Stephen Welch. Artificial neural network to predict traction performance of autonomous ground vehicle on a sloped soil bin and uncertainty analysis. *Computers and Electronics in Agriculture*, 196:106867, 2022. 6
- [2] Ayan Banerjee, Palaiahnakote Shivakumara, Saumik Bhat-tacharya, Umapada Pal, and Cheng-Lin Liu. An end-to-end model for multi-view scene text recognition. *Pattern Recognition*, 149:110206, 2024. 2
- [3] Karol B Barragan-Fonseca, Marcel Dicke, and Joop JA van Loon. Nutritional value of the black soldier fly (hermetia illucens L.) and its suitability as animal feed—a review. *Journal of Insects as Food and Feed*, 3(2):105–120, 2017. 1
- [4] Yan Cang, Hengxiang He, and Yulong Qiao. An intelligent pig weights estimate method based on deep learning in sow stall environments. *IEEE Access*, 7:164867–164875, 2019. 2
- [5] Wenbai Chen, Mengchen Liu, ChunJiang Zhao, Xingxu Li, and Yiqun Wang. Mtd-yolo: Multi-task deep convolutional neural network for cherry tomato fruit bunch maturity detection. *Computers and Electronics in Agriculture*, 216:108533, 2024. 2
- [6] Zheng Chu and Jiong Yu. An end-to-end model for rice yield prediction using deep learning fusion. *Computers and Electronics in Agriculture*, 174:105471, 2020. 2
- [7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995. 4
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2
- [9] Jonathan A Foley, Navin Ramankutty, Kate A Brauman, Emily S Cassidy, James S Gerber, Matt Johnston, Nathaniel D Mueller, Christine O’Connell, Deepak K Ray, Paul C West, et al. Solutions for a cultivated planet. *Nature*, 478(7369):337–342, 2011. 1
- [10] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. 4
- [11] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2
- [12] Thorben Grau, Andreas Vilcinskas, and Gerrit Joop. Sustainable farming of the mealworm tenebrio molitor for the production of food and feed. *Zeitschrift für Naturforschung C*, 72(9-10):337–349, 2017. 1
- [13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1
- [14] Glenn Jocher, K Nishimura, T Mineeva, and R Vilariño. yolov5. *Code repository <https://github.com/ultralytics/yolov5>*, page 9, 2020. 1, 2
- [15] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. 1
- [16] Paweł Majewski, Piotr Zapotoczny, Piotr Lampa, Robert Burduk, and Jacek Reiner. Multipurpose monitoring system for edible insect breeding based on machine learning. *Scientific Reports*, 12(1):7892, 2022. 1, 2, 6
- [17] Paweł Majewski, Piotr Lampa, Robert Burduk, and Jacek Reiner. Mixing augmentation and knowledge-based techniques in unsupervised domain adaptation for segmentation of edible insect states. In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pages 380–387. INSTICC, SciTePress, 2023. 3
- [18] Paweł Majewski, Mariusz Mrzygłód, Piotr Lampa, Robert Burduk, and Jacek Reiner. Monitoring the growth of insect larvae using a regression convolutional neural network and knowledge transfer. *Engineering Applications of Artificial Intelligence*, 127:107358, 2024. 1, 3, 4, 7
- [19] Puneet Mishra and Dário Passos. Multi-output 1-dimensional convolutional neural networks for simultaneous prediction of different traits of fruit based on near-infrared spectroscopy. *Postharvest Biology and Technology*, 183:111741, 2022. 2
- [20] Sarah Nawoya, Frank Ssemakula, Roseline Akol, Quentin Geissmann, Henrik Karstoft, Kim Bjerge, Cosmas Mwikirize, Andrew Katumba, and Grum Gebreyesus. Computer vision and deep learning in insects for food and feed production: A review. *Computers and Electronics in Agriculture*, 216:108503, 2024. 1
- [21] Dennis GAB Ooninx and Imke JM De Boer. Environmental impact of the production of mealworms as a protein source for humans—a life cycle assessment. *PLoS one*, 7(12):e51145, 2012. 1
- [22] JA Cortes Ortiz, A Torres Ruiz, JA Morales-Ramos, M Thomas, MG Rojas, JK Tomberlin, L Yi, R Han, L Giroud, and RL Jullien. Insect mass production technologies. In *Insects as sustainable food ingredients*, pages 153–201. Elsevier, 2016. 1
- [23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 1
- [24] Arnold Van Huis, Joost Van Itterbeeck, Harmke Klunder, Esther Mertens, Afton Halloran, Giulia Muir, and Paul Van-tomme. *Edible insects: future prospects for food and feed security*. Number 171. Food and agriculture organization of the United Nations, 2013. 1
- [25] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 13029–13038, 2021. 4
- [26] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7464–7475, 2023. 2
- [27] Yihan Wang, Xinglong Deng, Jianqiao Luo, Bailin Li, and Shide Xiao. Cross-task feature enhancement strategy in

- multi-task learning for harvesting sichuan pepper. *Computers and Electronics in Agriculture*, 207:107726, 2023. [2](#)
- [28] Dong Wu, Man-Wen Liao, Wei-Tian Zhang, Xing-Gang Wang, Xiang Bai, Wen-Qing Cheng, and Wen-Yu Liu. Yolop: You only look once for panoptic driving perception. *Machine Intelligence Research*, 19(6):550–562, 2022. [2](#)
- [29] Shuai Xiang, Siyu Wang, Mei Xu, Wenyan Wang, and Weiguo Liu. Yolo pod: a fast and accurate multi-task model for dense soybean pod counting. *Plant methods*, 19(1):8, 2023. [2](#)
- [30] Fan Zhang, Jin Gao, Chaoyu Song, Hang Zhou, Kunlin Zou, Jinyi Xie, Ting Yuan, and Junxiong Zhang. Tpmv2: An end-to-end tomato pose method based on 3d key points detection. *Computers and Electronics in Agriculture*, 210:107878, 2023. [2](#)
- [31] Jianlong Zhang, Yanrong Zhuang, Hengyi Ji, and Guanghui Teng. Pig weight and body size estimation using a multiple output regression convolutional neural network: A fast and fully automatic method. *Sensors*, 21(9):3218, 2021. [2](#)
- [32] Lei Zhang, Haisheng Li, Ruijun Liu, Xiaochuan Wang, and Xiaoqun Wu. Weakly supervised end-to-end domain adaptation for person re-identification. *Computers and Electrical Engineering*, 113:109055, 2024. [2](#)



## **4.7 Phenotyping with dynamic characteristics determination for Tenebrio Molitor beetles in selective breeding using re-identification**

**Authors:** Paweł Majewski, Piotr Lampa, Robert Burduk, Jacek Reiner, and Ta-Te Lin

**Publication status:** submitted to Engineering Applications of Artificial Intelligence

**Type of publication:** journal paper

**Lead Author:** Yes

**Corresponding Author:** Yes

**Percentage contribution:** 60%

**CRedit:** Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing–original draft preparation, Funding acquisition

# Phenotyping with dynamic characteristics determination for *Tenebrio Molitor* beetles in selective breeding using re-identification

Paweł Majewski<sup>a,\*</sup>, Piotr Lampa<sup>b</sup>, Robert Burduk<sup>a</sup>, Jacek Reiner<sup>b</sup> and Ta-Te Lin<sup>c</sup>

<sup>a</sup>*Faculty of Information and Communication Technology, Wrocław University of Science and Technology, 27 Wybrzeże Wyspiańskiego st., 50-370 Wrocław, Poland*

<sup>b</sup>*Faculty of Mechanical Engineering, Wrocław University of Science and Technology, 27 Wybrzeże Wyspiańskiego st., 50-370 Wrocław, Poland*

<sup>c</sup>*Department of Biomechatronics Engineering, National Taiwan University, No. 1, Roosevelt Rd., Sec. 4, Taipei Taiwan, ROC*

---

## ARTICLE INFO

**Keywords:**  
phenotyping  
dynamic  
re-identification  
domain shift  
self-supervised  
*Tenebrio Molitor*

## ABSTRACT

Selective breeding studies for edible insects enable the development of high-quality rearing input, resulting in higher daily larval mass gains. Currently, the selection of the best beetles for reproduction is based on static characteristics, i.e. weight and geometric dimensions that are insufficient to characterize individuals fully. In this research, we proposed a phenotyping method with dynamic characteristics determination for *Tenebrio Molitor* beetles using re-identification. The proposed procedure consisted of two stages: training and testing. During the training stage, the beetles were isolated in individual stations, which allowed the development of a training set for a re-identification model without manually labeling samples. During the test stage, the beetles were free to move and interact. Re-identification in the test stage was carried out based on the segmented abdomen of beetles, and a physical tag located on the head of the beetle served as ground truth. For the best re-identification model developed, a precision at 1 (P@1) of 0.807 was obtained when analyzing 80 individuals. The P@1 value was further increased to 0.853 through the proposed method of reducing the domain shift effect. The study also proposed a strategy for the initial selection of beetles for phenotyping, which made it possible to increase the number of simultaneously analyzed beetles significantly. The ablation studies showed the high role of color-related features for re-identification and confirmed the solution's reliability. The high quantitative results obtained confirm the implementation potential of the proposed phenotyping method.

---


## 1. Introduction

Mealworms (*Tenebrio Molitor*) are one of the most popular representatives of edible insects used for food and feed production (Tang et al. (2019)). Due to their rich mineral composition (especially protein and fat) (Costa et al. (2020)) and sustainable farming (low global warming potential, low water footprint, and low space utilization for a certain amount of final product) (Grau et al. (2017)), more and more entities in European Union (EU) countries are interested in farming mealworms (Mancini et al. (2022)).

In most cases, achieving cost-effectiveness in the farming of edible insects requires the operation on a large scale and the automation of farming activities, including feed preparation,

---

\*Corresponding author

 pawel.majewski@pwr.edu.pl (P. Majewski)

ORCID(s): 0000-0001-5076-9107 (P. Majewski)

feeding, sorting, and post-rearing harvesting (Heckmann et al. (2018)). Manual monitoring of large-scale farming is impossible, necessitating the use of tools that digitalize farming (Neethirajan and Kemp (2021)), most often using machine vision systems, computer vision (CV) methods and machine learning (ML) models. In the literature, we can find few works on monitoring systems for the farming of edible insects. Majewski et al. (2022) proposed a multipurpose system for monitoring the rearing of the mealworms, which included counting growth stages (larvae, pupae, beetles), detection of anomalies (dead larvae, pests in the form of *Alphitobius diaperinus Panzer*), estimation of the amount of chitinous moults and food residues in the rearing box, and basic phenotyping (dimensioning) of larvae during growth. Majewski et al. (2024) focused on developing and improving the phenotyping part of the system by increasing the robustness of methods to dense scenes and significant size differentiation of larvae during rearing and on reducing processing time. The mentioned and present works in the literature concerned only the rearing period, that is, the period of farming from small larvae as input (length of about 10 mm) to mature larvae as output (length of about 30 mm), which can be further processed.

An essential element determining the rearing efficiency of mealworm larvae is the quality of the input material, which influences, e.g., the length of rearing and the daily mass gain of larvae. Conducting selective breeding studies is aimed at obtaining high-quality input material. Researchers have used selective breeding studies for mealworms to increase pupal size, growth rate, fecundity, the efficiency of food conversion (Morales-Ramos et al. (2019)), immunity (Armitage and Siva-Jothy (2005)), and to color modification (Song et al. (2022)). One of the critical elements in selective breeding studies is the selection of adult individuals (beetles) for further reproduction. The problem arises already at the stage of determining the sex of an individual, which requires time-consuming manual examination by a specialist for each individual separately (Bhattacharya et al. (1970)). On the other hand, selecting the best individual from a group of beetles of the same sex is also not obvious. The often used "bigger is better" approach (Morales-Ramos et al. (2019)) and basing only on static characteristics may not be the most optimal solution here. Morales-Ramos et al. (2019) observed that populations with increased size and biomass productivity may have lower larval survival. Undoubtedly, there is a need to propose a procedure and metrics to objectively determine the reproductive quality of beetles for selective breeding studies. In response to the problems described, this research proposes a procedure of phenotyping with dynamic characteristics determination for *Tenebrio Molitor* beetles in selective breeding studies, which is based on observations of the mobility of individuals and their interactions (e.g., mating pattern detection).

The beetles phenotyping mentioned in the previous paragraph requires re-identification (recognizing the same individual on successive frames), which is a complex task considering the high similarity in the appearance of beetles. The problem of re-identification is a relatively common issue addressed by researchers in the context of humans (Gong et al. (2011)), livestock e.g. dairy cows (Wang et al. (2004); Chen et al. (2021)) and pigs (Wang et al. (2022)), and wildlife (Schneider et al. (2019)), e.g. tigers (Li et al. (2019)), zebras (Lahiri et al. (2011)) and whale sharks (Arzoumanian et al. (2005)).

In the literature, we can also find the first works on the re-identification of insects. In (Murali et al. (2019)), the re-identification problem of fruit flies was addressed. In this paper, the re-identification was posed as a classification problem using the ResNet18 feature extractor with a softmax output layer composed of a certain number of output units corresponding to the number of considered individuals. The researchers emphasized the importance of the domain shift phenomenon on re-identification accuracy and proposed adaptation methods, including (1) training the model using images from two days instead of one, (2) applying augmentation techniques such as random cropping, random masking, and (3) using a domain adversarial neural network (Ganin et al. (2016)). It should be noted that the dataset for the experiments

was developed by acquiring images separately for each individual under laboratory conditions. The dataset did not include individuals during interactions under real conditions, which is very important from an application point of view. For more extensive evaluation, ablation studies should also have been conducted (as in (Borlinghaus et al. (2023))), e.g. in the form of re-identification based only on the segmented insects' bodies, which would have eliminated the potential influence of background on re-identification and enabled identify the most important features of the insect's appearance used for re-identification.

Tausch et al. (2020) proposed a dataset for the re-identification of bumblebees, and Borlinghaus et al. (2023) proposed a re-identification method based on this dataset. As the ResNet18 model was trained with metric learning and triplet loss function, the 128-dimensional embeddings obtained for individual images were then used for re-identification. In the paper described above, ablation studies relevant to the problem were conducted, showing that excluding the background during training and basing only on the segmented body of the bumblebee results in a significant reduction in re-identification accuracy, which can be explained by the high importance of legs and wings for re-identification or the undesirable background features. The authors also emphasized the important role of shape features for re-identification.

In the work by Chan et al. (2022), they focused on re-identifying honey bees based on images of the abdomen. The individual's label was a tag located on the bee's head, which was excluded from the training/test images. Image acquisition took place in a specially designed station that allowed individuals to interact with each other. A custom architecture based on convolutional neural networks, including ResNet blocks, was used as a model. The training was carried out using metric learning with semi-hard triplet loss. The reported work proposed a self-supervised approach for acquiring training images through tracking methods. The researchers emphasized the importance of augmentation techniques (color distortion, color drop, Gaussian blur and random cropping) and the number of training images on re-identification accuracy.

Based on the works mentioned in the field of insect re-identification, we can draw the following conclusions: (1) the re-identification problem should be set in the context of a specific application problem, and the validation should be carried out under real conditions, e.g., where there is the interaction between individuals, (2) for a deeper understanding of the performance of the re-identification model, additional experiments (e.g., ablation studies) are required to exclude the situation where the model prediction is adversely affected by inappropriate features, e.g. from the background, (3) the problem of domain shift occurs between the training and test stages and can be reduced by self-supervised adaptation methods, and (4) a physical tag is necessary for a fair evaluation of re-identification, as humans do not have a high ability to re-identify animals to annotate them manually.

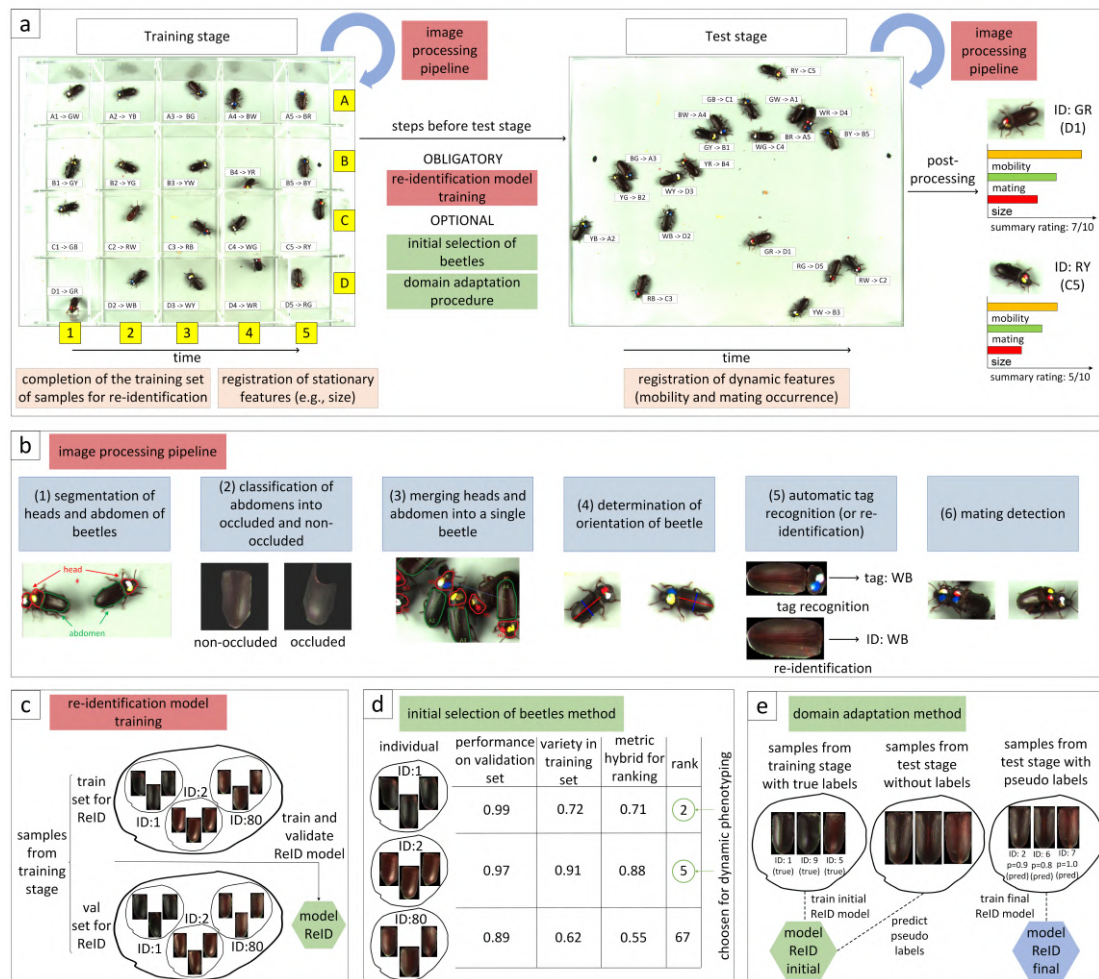
Considering the above, we propose a method of phenotyping *Tenebrio Molitor* beetles in selective breeding studies based on re-identification and mating detection. The study proposed (1) an automatic completion method for a re-identification training set with no possibility of mislabeling, (2) a fair evaluation for re-identification performance where ground truth is determined based on a physical tag, (3) a self-supervised method for reducing the effect of domain shift on re-identification accuracy, and (4) ablation studies to identify the most important features used for re-identification. In addition, an original method was developed for the initial selection of beetles for re-identification using the proposed hybrid metric.

## 2. Materials and Methods

### 2.1. Definition of the problem

The problem addressed in this publication concerned a method for the non-invasive determination of highly informative characteristics for individual beetles, which can indirectly determine the reproductive value of beetles and be the basis for selecting individuals in selection breeding studies. Among the characteristics to be calculated were those of a static and dynamic nature.

For static characteristics, recording characteristics could be carried out on single frames, which was not a challenging research task. In the case of dynamic characteristics, re-identification of individuals in the test stage was required, necessitating prior training of the re-identification model on a prepared set of training samples. The proposed static and dynamic characteristics are described in more detail in the 2.4 section. The developed solution should be characterized by high performance, robustness and an adequate level of interpretability, which was ensured by conducting subsequent dedicated experiments. The idea scheme for the proposed solution is shown in Figure 1, and the next steps of this solution are discussed in the following sections.



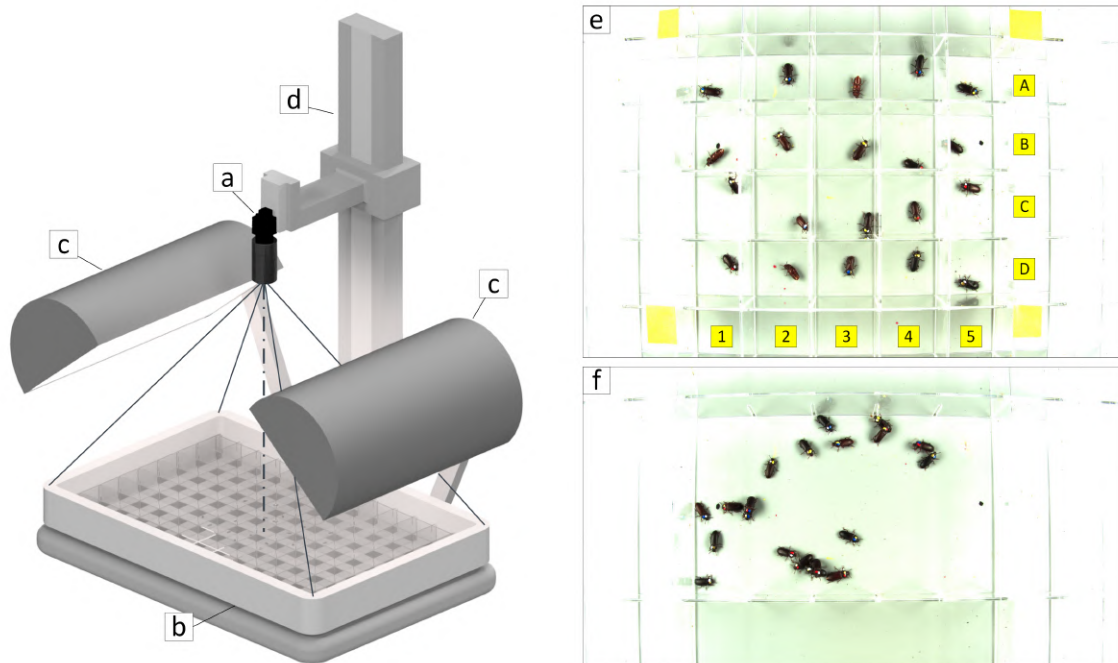
**Figure 1:** Idea scheme for the proposed solution: (a) phenotyping procedure, (b) image processing pipeline, (c) re-identification model training, (d) initial selection of beetles method and (e) domain adaptation method.

## 2.2. Data acquisition

In the present studies, images of beetles were obtained using a prepared data acquisition station (Figure 2). A Phoenix PHX120S-CC color camera (LUCID Vision Labs, Richmond, Canada) with a resolution of 4096 x 3000 pixels and a 12 mm lens was applied. The camera was placed 490 mm above the rearing box, in which a transparent grating made of acrylic glass was placed to form separate cubic spaces for individual beetles. The smooth surface of the acrylic glass and the height of the plates of 40 mm prevented the beetles from moving outside their space. The setup was illuminated by fluorescent lamps in diffusing reflectors. The example raw images in Figure 2 may appear overexposed, but this approach was intentional, in order to highlight the features of the abdomens and heads of naturally dark beetles.



Imaging was carried out in two stages. For the training stage, where beetles were kept in separated spaces (Figure 2e), images were captured every 5 seconds, for 20 minutes. This frequency guaranteed a noticeable change in the position of the insects between acquisitions. For the test stage, some of the grating elements were removed to create a common space for all beetles (Figure 2f). This time images were captured for 15 minutes, also every 5 seconds.



**Figure 2:** Data acquisition station: (a) camera with lens, (b) rearing box with a transparent grating of acrylic glass forming partitions for individual beetles (c) illuminators, (d) camera mount; and raw images of the experimental setup with separated (e) and common (f) space for beetles.

### 2.3. Dataset for re-identification

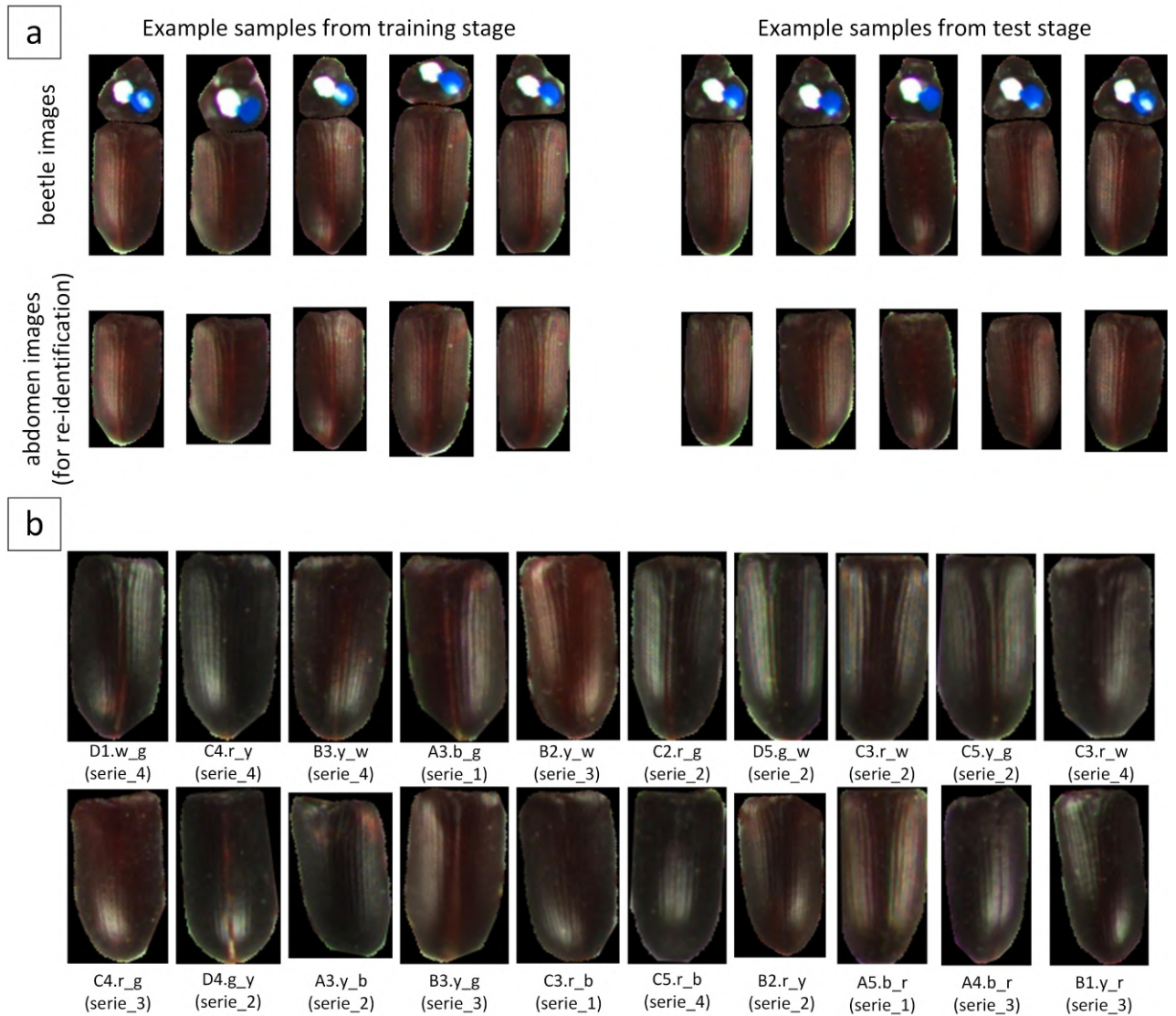
The development of the re-identification dataset was mostly automated by sequentially: (1) segmenting the heads and abdomens of the beetles using the YOLOv8-seg model (see section 2.5), (2) filtering out occluded abdomens using a classification model based on the pre-trained backbone (see section 2.6), (3) merging detected heads and non-occluded abdomens into a single beetle (see section 2.7), (4) determining beetle orientation (see section 2.8), and (5) automatic tag recognition (see section 2.9). The few errors were eliminated by manually checking the samples in the automatically completed training and test sets.

The images used for training and testing the re-identification model were abdomen images (without background) whose rotation was normalized using a pre-determined orientation. Example samples from the re-identification dataset are shown in Figure 3.

An analogous dataset containing images of whole beetles (including the head containing the tag) was also developed for research purposes, allowing the determination of an upper baseline for re-identification.

The procedure for beetles phenotyping (see Figure 1a) for obtaining samples for training and testing re-identification models was carried out identically to that for beetles without a tag. A total of 4 separate phenotyping procedures (training stage + test stage) were carried out each time for 20 individuals, making a total of 80 evaluated individuals. A summary of the developed re-identification dataset is shown in Table 1.

## Phenotyping of *Tenebrio Molitor* beetles



**Figure 3:** Example samples from the re-identification dataset: (a) samples from the training and test stage for the chosen beetle and (b) samples for different beetles.

**Table 1**

Summary of the re-identification dataset.

stage	no. samples (per individual)			all samples
	mean	min	max	
training	171	33	253	13677
test	110	35	171	8775

### 2.4. Determination of stationary and dynamic characteristics of beetles

Phenotyping of individual beetles involves the determination of static characteristics and dynamic characteristics. Dynamic characteristics were determined from observations during the test stage.

#### 2.4.1. Stationary characteristics of beetles

As part of this research, the following static characteristics of beetles were determined: (S1) size of the beetle's head, (S2) size of the beetle's abdomen, (S3) length of the major axis for the

beetle abdomen, and (S4) length of the minor axis for the beetle abdomen. Features S1-S2 were calculated as binary mask areas, while features S3-S4 were calculated as distances between characteristic points. The procedure for determining the characteristic points is described in detail in section 2.8.

#### **2.4.2. Dynamic characteristics of beetles**

In these studies, the following were defined as dynamic characteristics: (D1) mobility and (D2) mating frequency. Mobility was measured by recording the successive movements of re-identified beetles during the test stage. The proposed mating detection method is described in section 2.13.

### **2.5. Segmentation of the heads and abdomen of beetles**

The motivation for segmenting the beetles' heads and abdomen was to separate these two body parts, which was related to placing a tag (ground truth for re-identification) on the beetles' heads. In this study, only images of the beetles' abdomen were used for re-identification.

To segment the heads and abdomen of beetles, an instance segmentation model had to be developed for these two proposed object classes (head, abdomen). The research used a YOLOv8(-seg) (Jocher et al. (2023)) model adapted to the segmentation task. Various proposed architectures for this model (n, s, m, l, x versions) were evaluated to find a trade-off between performance and inference time. Inference for instance segmentation task was performed on 640x640 tiles - parts of whole images acquired during acquisition. The dataset for the beetle head and abdomen segmentation task consisted of 889 head annotations and 931 abdomen annotations. The dataset included images from the training and test stages of phenotyping, images representing occluded and non-occluded body parts, and heads with and without tags. Details on evaluating this image processing part are presented in the section 2.15.

### **2.6. Classification of detected abdomens into occluded and non-occluded**

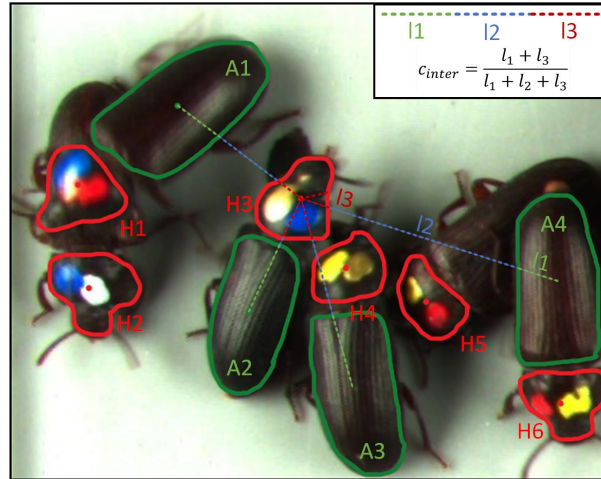
In this research, it was assumed that re-identification was to be performed only for non-occluded abdomen. It was decided that re-identification in the presence of occlusion is a complex topic, and it is worth dedicating a separate study to this problem.

To distinguish occluded from non-occluded abdomen, a classification model was developed based on pre-trained on ImageNet (Krizhevsky et al. (2017)) backbones: ResNet18, ResNet50, ResNet101 (He et al. (2016)), EfficientNet-b0, EfficientNet-b4 (Tan and Le (2019)), and MobileNetv2 (Sandler et al. (2018))). Three fully connected (FC) hidden layers were also added to the backbone, where the number of output features determined the number of neurons for a specific backbone, i.e. for 512 (ResNet18), the following number of neurons in FC layers was used: [256, 128, 64], for 1028 (MobileNetv2, EfficientNet-b0): [512, 256, 128] and for 1792 (EfficientNet-b4) and 2048 (ResNet50, ResNet101): [1024, 512, 256]. Inference for the classification model was performed using 128x128 images. Images of the abdomen after segmentation and after rotation normalization were placed on a black background in the center of the image. No re-sizing was done for the abdomen images since the size of the segmented object was important information in the context of the occlusion classification problem. The dataset for the occluded/non-occluded abdomen classification task contained 840 samples of the non-occluded abdomen and 123 samples of the occluded abdomen. Before training, oversampling was performed for the minority class (occluded abdomen) and normalization of the R, G, and B channels using the calculated means and standard deviations of the intensities for each channel. The following classification model training parameters were used: optimizer Adam, binary cross-entropy loss, number of epochs: 200, learning rate:  $10^{-5}$ , batch size: [32, 64, 128] depending on the model. For training, all weights were unfrozen (all parameters were trainable). Details regarding the quantitative evaluation of the classification of the abdomen into occluded and non-occluded are placed in the section 2.15.

## 2.7. Merging the detected heads and abdomen into a single beetle

This section describes the procedure for merging the detected beetle heads and abdomen into a single beetle. This is an important processing step as it allows further determination of beetle orientation and automatic labeling of the beetle abdomen for re-identification.

The procedure for merging the detected beetle heads and abdomen into a single beetle is shown in Figure 4.



**Figure 4:** Procedure for merging detected beetle heads and abdomen into a single beetle.

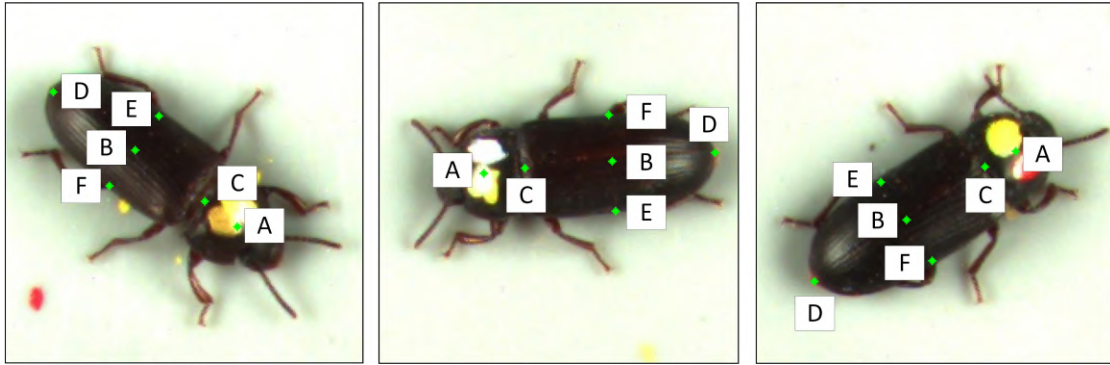
Suppose that after segmentation of the heads and abdomen of the beetles and after selecting the non-occluded abdomen, we have two sets:  $H$  and  $A$ , where  $H$  is the set of all detected beetle heads and  $A$  is the set of all detected non-occluded beetle abdomen. Each element from the set  $H$  and  $A$  is associated with a binary mask and the midpoint of the binary mask. We combine the selected beetle head  $H_i$  with the corresponding abdomen  $A_j$ . For each pair  $(H_i, A_j)$ , where  $i = \text{const}$ ,  $j = 1, \dots, m$ , a section connecting the midpoints of the binary masks and an intersection coefficient was determined, specifying the part of the determined section contained within the binary masks region. Among  $A_j$  for different  $j$ , we choose the element for which the intersection coefficient is the highest. For a pair  $(H_i, A_j)$  to be considered valid, the value of the intersection coefficient  $c_{inter}$  must be sufficiently high (greater than the threshold value  $c_{inter}^{thresh}$ ). If this condition is not met, the correct abdomen was not detected or was covered. The threshold value  $c_{inter}^{thresh}$  was determined by analyzing the values of the intersection coefficient  $c_{inter}$  for different correct head-abdomen pairs. The value of  $c_{inter}^{thresh}$  was determined based on the minimum value of  $c_{inter}$  among the analyzed pairs and the selected offset.

## 2.8. Determination of the orientation of the beetle

Determination of the orientation of the beetle is an important issue in the problems addressed, as it allows (1) registration of static characteristics, e.g., the length of the major and minor axes, (2) reading the tag located on the beetle's head and (3) normalizing the beetle's rotation before re-identification. In the conducted research, the problem of determining the orientation of the beetle was posed as determining the position of six characteristic points, i.e.  $A$  - the center of the head,  $B$  - the center of the abdomen,  $C$  - the upper border point of the abdomen,  $D$  - the lower border point of the abdomen,  $E$  - the left border point of the abdomen and  $F$  - the right border point of the abdomen. The characteristic points for the example beetles are shown in Figure 5.

The characteristic points  $A$  and  $B$  were determined as the centers of the binary masks of the head and abdomen. Other characteristic points ( $C$ ,  $D$ ,  $E$ , and  $F$ ) were selected from among the points belonging to the contour and under the following assumptions:





**Figure 5:** The characteristic points for example beetle.

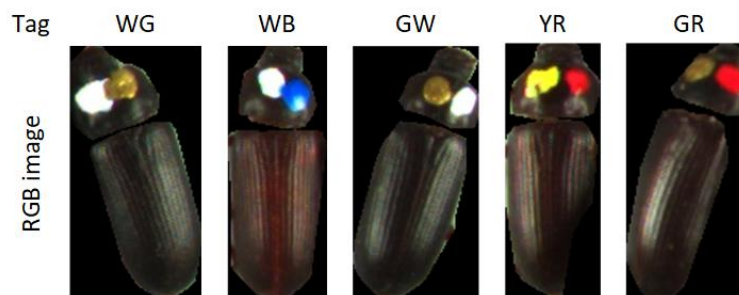
1. the angle between the lines  $l_{AB}$  and  $l_{BC}$  should be relatively small (smaller than the threshold angle  $\alpha_{T1}$ ),
2. the angle  $\angle CBE$  should be approximately right (deviation from the right angle smaller than the threshold angle  $\alpha_{T2}$ ),
3. the ratio  $|BC|/|BE|$  should be as large as possible when conditions 1. and 2. are met, where  $|BC|$  and  $|BE|$  are the lengths of the sections between the defined characteristic points,
4. the characteristic points  $D$  and  $F$  are determined after  $C$  and  $E$  as the contour points closest to line  $l_{BC}$  (for point  $D$ ) and line  $l_{BE}$  (for point  $F$ ).

To satisfy condition 3., a certain number of pairs of contour points were checked and the pair for which the ratio  $|BC|/|BE|$  had the largest value was selected. In order to fine-tune the  $\alpha_{T1}$  and  $\alpha_{T2}$  parameters, the major axes for 50 beetle images were marked manually. Details regarding the evaluation of beetle orientation determination are described in the section 2.15.

The length of the major axis for the beetle abdomen (S3) was determined as the length of the  $|CD|$  section, and analogously, the length of the minor axis for the beetle abdomen (S4) was determined as the length of the  $|EF|$  section.

### 2.9. Automatic tag recognition

Tags on the beetles' heads enable them to be identified. In order to automatically identify the beetles, a method had to be developed to read the two-element tag located on the beetle's head. There could be five types of colors in the tag: red (R), blue (B), yellow (Y), gold (G) and white (W), giving a total of 20 combinations for unique tags, e.g. RY (red and yellow). The same colors were not repeated in the tag. Examples of beetle head tags are shown in Figure 6.

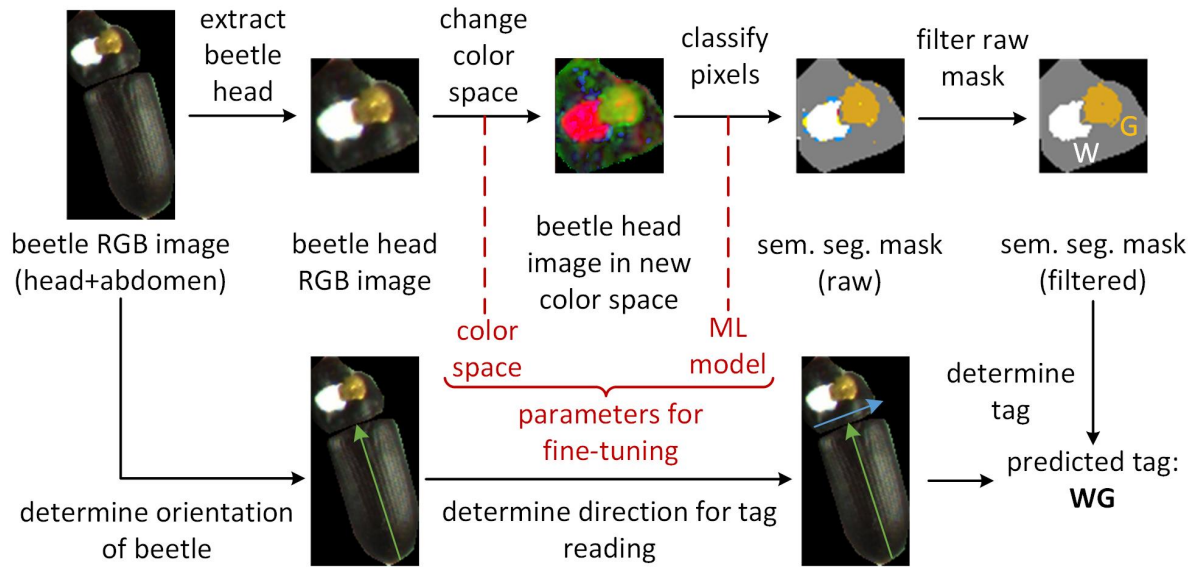


**Figure 6:** Examples of beetle head tags.

A method based on semantic segmentation of the beetle head image was proposed to facilitate reading the beetle head tag. The procedure for automatic tag reading is shown in Figure 7.



## Phenotyping of *Tenebrio Molitor* beetles



**Figure 7:** Procedure for automatic reading of the tag located on the beetle's head.

The classes defined for the semantic segmentation represented the colors used for tagging (R, B, Y, G and W) and the background (BG), e.g. fragments of the beetle head. Image fragments representing the defined classes were extracted to train the pixel classification model. The number of extracted areas is summarized in Table 2. Each pixel in the annotated area was one training sample.

**Table 2**

Number of annotated areas for the semantic segmentation of beetle head image.

class name	R	B	Y	G	W	BG
no. of annotations	29	29	34	21	31	41

Classical machine learning models were used to perform pixel classification: Logistic Regression (LogReg), Linear Discriminant Analysis (LDA), SVM with linear kernel (SVM linear) and SVM with radial basis function kernel (SVM rbf). In the original version, the beetle head images were in RGB space, so each training sample was characterized by three-pixel intensities: R, G and B. In one experiment, other color spaces (HSV, Lab, Luv, YCrCb) were also tested to improve classification accuracy. Each (machine learning model, color space) combination was tested when fine-tuning the settings. Finally, the best combination of settings was chosen to read the tags. Details on the evaluation of semantic segmentation for automatic tag recognition are included in section 2.15.

After semantic segmentation, the resulting mask was filtered to separate the two main colors belonging to the tag from the noise. The two largest clusters formed by pixels representing one of the five defined colors were used as the predicted tag colors. Each tag contained two different colors. The order of the colors was important. The study assumed that the tag had to be read from left to right, looking from the abdomen to the head.

For the evaluation of the method for tag reading, 111 tagged beetles were selected. A tag manually read by the user was used as ground truth. Details on the evaluation of the entire process for automatic tag recognition are described in section 2.15.

### 2.10. Development of a model for the re-identification of beetles for the test stage

The beetle re-identification model was trained using images from the dataset described in section 2.3, which are images of beetle abdomens. Metric learning techniques were used for training to obtain problem-oriented embeddings, i.e., to distinguish individual beetles from each other. The selection of the best model and training parameters was conducted in stages. In the first stage, the best backbone was selected from among the pre-trained architectures: ResNet18, ResNet50, ResNet101 (He et al. (2016)), EfficientNet-b0, EfficientNet-b4 (Tan and Le (2019)), and MobileNetv2 (Sandler et al. (2018)). The model in the first stage consisted of a specific backbone and one (default) FC (fully connected) layer consisting of 512 neurons. Triplet Margin Loss and Triplet Margin Miner were used to train the re-identification model with a margin value of 0.2 for both loss and miner. The selected default distance metric was cosine distance, and the triplets type was semihard. The triplets approach has been used frequently in other re-identification-related studies (Chen et al. (2021); Borlinghaus et al. (2023); Chan et al. (2022)).

The triplets mentioned above consist of a sample called anchor, one positive sample (represents the same individual as the anchor) and one negative sample. Using the chosen feature extractor, we calculate embeddings for these three samples. Let's specify  $d_{ap}$  as the distance in feature space (i.e. cosine distance) between the anchor and the positive sample and as  $d_{an}$  the distance between the anchor and the negative sample. The formula for Triplet Margin Loss for the  $i$ -th triplet is as follows:

$$L_{triplet} = \max(d_{ap}^i - d_{an}^i + margin, 0) \quad (1)$$

where *margin* is the value of the margin.

The result of the first stage was the selection of the best backbone. In the second stage, different FC layer structures were tested: [] (no hidden layers), [1024], [512, 256], [1024, 512], [512, 256, 128], [1024, 512, 256]. The structure with the best results was used in the next stages. In the third stage, various losses (different from the default Triplet Margin Loss) were tested, i.e. Circle Loss (Sun et al. (2020)), Generalized Lifted Structure Loss (Hermans et al. (2017)), Multi-Similarity Loss (Wang et al. (2019)), Proxy-NCA Loss (Movshovitz-Attias et al. (2017)) and FastAP Loss (Cakir et al. (2019)). In stage four, parameters were fine-tuned for best loss and miner. Details regarding the settings in the subsequent stages of model parameter selection and training are summarized in Table 12, which can be found in the Appendix. The study used implementations of methods from the metric-learning library (Musgrave et al. (2020b)).

The study also proposed additional experiments to determine the re-identification model's two lower and upper baselines. The first lower baseline was determined by the approach when the weights in the backbone were frozen. The second lower baseline was related to relying on handcrafted features, a set of standard features calculated for beetle abdomen images. A total of 39 handcrafted features related to shape (Hu moments), texture (Haralick features), intensity (mean values for L, a, b channels) and geometric parameters (area, lengths of major and minor axis of ellipse, eccentricity, solidity) were used. Implementations of methods from the scikit-image library (Van der Walt et al. (2014)) were used to calculate handcrafted features. The upper baseline was an approach where images of the abdomen with the head (where the tag was located) were used for training while unfreezing all model weights.

Inference of the re-identification model consisted of determining beetle IDs for specific samples (so-called query samples) based on samples collected in the training stage (so-called reference samples). Each sample represented one point in a multidimensional feature space. Inference was based on assigning an ID for the query based on the nearest reference sample.

Details regarding the evaluation of the re-identification model are provided in the section 2.15.

**Table 3**

Description of the transformations used for the ablation studies.

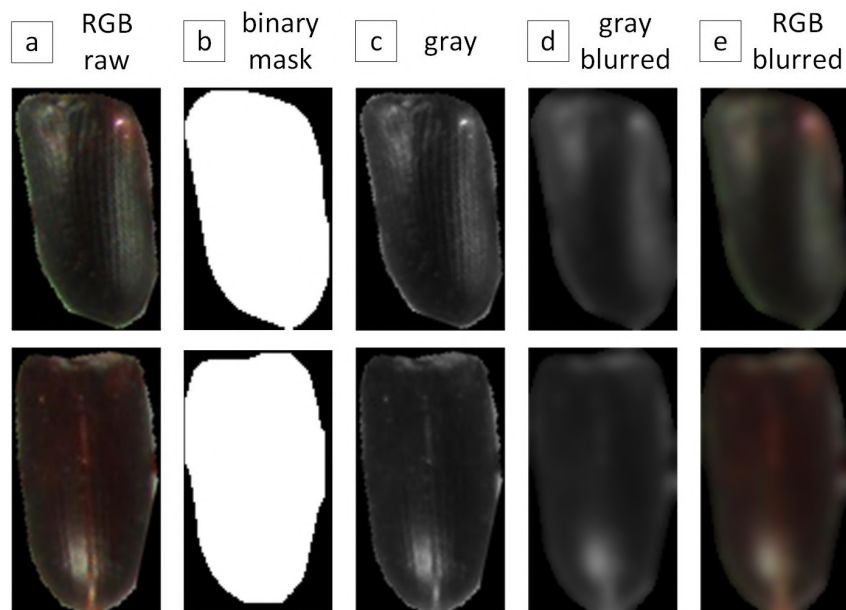
approach name	input image	output image	shape (S)	texture (T)	intensity (I)	color (C)
S	RGB raw	binary mask	+	-	-	-
S+T+I	RGB raw	grayscale	+	+	+	-
S+I	RGB raw	grayscale blurred	+	-	+	-
S+I+C	RGB raw	RGB blurred	+	-	+	+
S+T+I+C (baseline)	RGB raw	-	+	+	+	+

### 2.11. Ablation studies for re-identification

A comprehensive understanding of the inference of models for re-identification is very important, as it makes it possible to exclude situations in which the model inference is based on undesirable features (e.g., related to background, acquisition conditions), which can result from improperly performed acquisition or validation.

To increase our understanding of re-identification, in the case of *Tenebrio Molitor* beetles, ablation studies were proposed to answer the question of which type of feature (related to shape, texture, intensity, or color) is mainly responsible for the beetles' re-identification ability. Images from the training and test sets (hereafter referred to as RGB raw) were processed by the following transformations summarized in Table 3.

The purpose of the transformations described in Table 3 was to eliminate (or reduce) the influence of a selected group of features on re-identification, i.e. RGB-grayscale transformation resulted in the elimination of color-related features (only the intensity of pixels from one channel remains), Gaussian blurring resulted in the reduction of texture features on re-identification. For the new datasets developed (containing transformed images), training and validation were carried out according to the procedure described in the section 2.10. The results obtained after the evaluation were compared with the baseline (results obtained for a dataset containing raw RGB images). Example images after proposed transformations are shown in Figure 8.



**Figure 8:** Example images after proposed transformations for ablation studies: (a) RGB raw, (b) binary mask, (c) grayscale, (d) grayscale blurred, (e) RGB blurred.

## 2.12. Initial selection of beetles for re-identification for the test stage

It can be assumed that increasing the number of individuals evaluated simultaneously in the test stage should contribute to a decrease in re-identification performance, which is associated with a higher probability of finding pairs of individuals in the pool of evaluated beetles that are difficult to distinguish from each other. It is also worth noting that some individuals are easy to re-identify, for example, by having distinctive characteristics in appearance. Adding such individuals to the pool of simultaneously evaluated beetles will not significantly reduce re-identification performance.

This part of the study addressed the problem of initial selection of beetles for the test stage. The motivation for initial selection is the hypothesis that more individuals can be evaluated simultaneously, with a certain level of re-identification performance, compared to an approach where individuals for the test stage are selected randomly. Farmers have many beetles on the farm, so preparing more for initial selection should not be a problem.

The study identified two main factors influencing the difficulty of re-identifying a chosen individual during the test stage. The first was related to the individual's appearance and the ability to distinguish it from other individuals when training the model for re-identification in the training stage. When achieving relatively low performance on the validation set in the training stage, achieving high scores on the more challenging test set will be impossible in most cases. However, basing only on the received metrics values on the validation set for initial selection is not a fully justified approach. It should be noted that the examined beetles were characterized by different mobility in the training stage, which resulted in differences in the number of various views of the beetle. A higher variation in the training set for a selected beetle generally means a higher chance of robustness to the domain shift phenomenon occurring between the training and test stages. The beetles characterized by non-mobility were the source of nearly identical captures, resulting in the training of the re-identification model for a particular view (not individual). Taking these two factors into account, we can define a formula for a hybrid metric estimating the ease of re-identification of a given individual in the test stage:

$$\text{metric hybrid} = (\text{perf. on val set})^\alpha (\text{var. in train set})^\beta \quad (2)$$

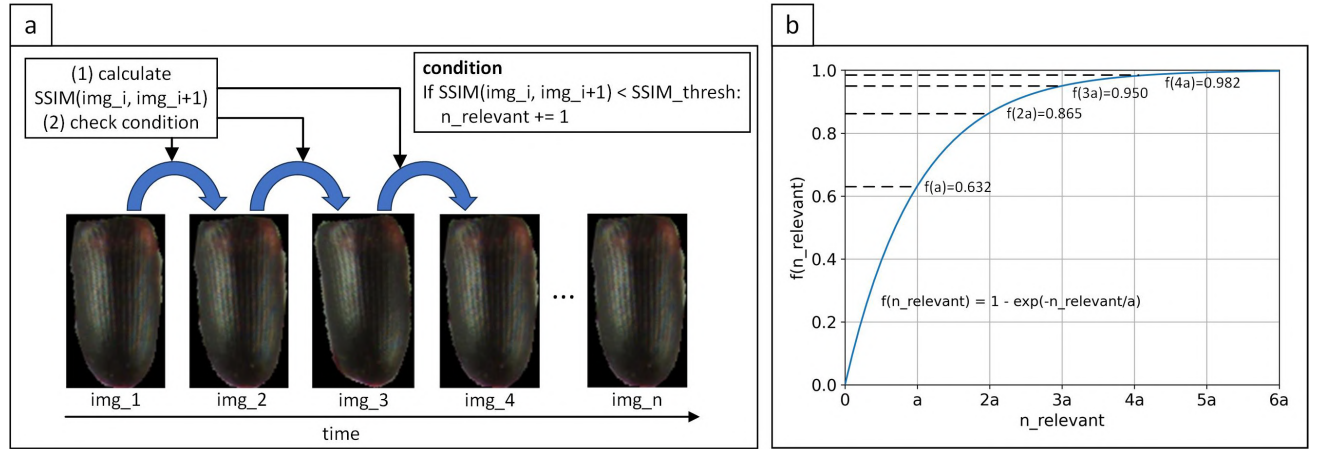
The coefficients  $\alpha$  and  $\beta$  represent the weights that determine the effect of each part on the ease of re-identification in the test stage. The higher value of the hybrid metric indicates a higher ease in re-identifying a given individual in the test stage. The ranking of the ease of re-identification of beetles, determined by the value of the hybrid metric, was the basis for an initial selection of beetles.

Let's assume that the value of the chosen metric (e.g., precision at 1) on the validation set was  $m_{ReID}^{val}$  and that the range of values for the given metric is 0-1. Let's also set a lower threshold for the given metric  $m_{ReID}^{thresh}$ , below which there is no validity in using the given individual for the test stage. Taking this into account, the part of the formula (2) related to performance on the validation set can be expressed by the equation:

$$(\text{perf. on val set})^\alpha = \left( \frac{m_{ReID}^{val} - m_{ReID}^{thresh}}{1 - m_{ReID}^{thresh}} \right)^\alpha \quad (3)$$

To increase reliability in determining performance on the validation set, the value of the metric  $m_{ReID}^{val}$  was the average value of the considered metric for re-identification based on the results obtained in cross-validation. The procedure for performing cross-validation for this problem is described in detail in the section 2.15. The equation (3) used min-max scaling for  $m_{ReID}^{val}$ .

The second part in the formula (2) determined the variety of samples in the training set. The procedure for quantifying the variety of samples in the training set is shown in Figure 9.



**Figure 9:** Procedure for quantifying the variety of samples in the training set: (a) counting relevant pairs of samples (significantly differing in appearance), (b) calculating the normalised coefficient based on the proposed exponential function.

The variety of samples in the training set was determined as follows. Suppose we have a series of beetle images from the training stage:  $img_1, img_2, \dots$  and  $img_n$  assuming that the images are sorted according to acquisition time. For each pair of neighboring samples, sequentially  $(img_1, img_2), (img_2, img_3), \dots, (img_{n-1}, img_n)$ , we calculate the structural similarity index (SSIM) (Wang et al. (2004)). Then we count the number of pairs of neighboring images  $i$  and  $i + 1$  for which the calculated  $SSIM(img_i, img_{i+1})$  is lower than  $SSIM_{thresh}$ . The value of  $SSIM_{thresh}$  indicates a significant change in appearance between neighboring images. In the case of no movement, the values of  $SSIM(img_i, img_{i+1})$  would be relatively high. Let  $n_{relevant}$  denote the number of pairs of neighboring images for which a significant change in appearance had occurred. The part of the formula (2) related to the variety of the training set can then be expressed by the following formula:

$$(var. \text{ in train set})^\beta = (1 - \exp(-n_{relevant}/a))^\beta \quad (4)$$

where  $a > 0$ . The formula (4) is derived from the assumption that at small  $n_{relevant}$ , increasing the value of parameter  $n_{relevant}$  would result in a greater gain in re-identification performance than a corresponding increase at large  $n_{relevant}$ . For the reason described, the formula (4) used an exponential dependence instead of a linear dependence. The introduction of the parameter  $a$  was intended to scale the parameter  $n_{relevant}$ , which facilitated further parameter fine-tuning and analysis of the function  $f(n_{relevant}) = (1 - \exp(-n_{relevant}/a))^\beta$ . For example, for  $n_{relevant} = 4a$ , the value of  $f(n_{relevant})$  was already very close to the maximum value (0.982) and further increasing  $n_{relevant}$  no longer contributed significantly to the value of  $f(n_{relevant})$ . The chart of the function  $f(n_{relevant})$  is shown in Figure 9b.

The determined values of the hybrid metric for individual beetles provided the basis for determining the ranking of beetles. Based on the ranking of beetles, a curve of re-identification performance versus the number of simultaneously evaluated beetles was determined. Each point on the chart was associated with a certain number of beetles less than or equal to the number of individuals used in the experiment. For example, for a point indicating  $n_{beetle}$  evaluated beetles,  $n_{beetle}$  best individuals were selected according to a designated ranking. A better ranking meant a greater area under the designed curve re-identification performance versus the number of

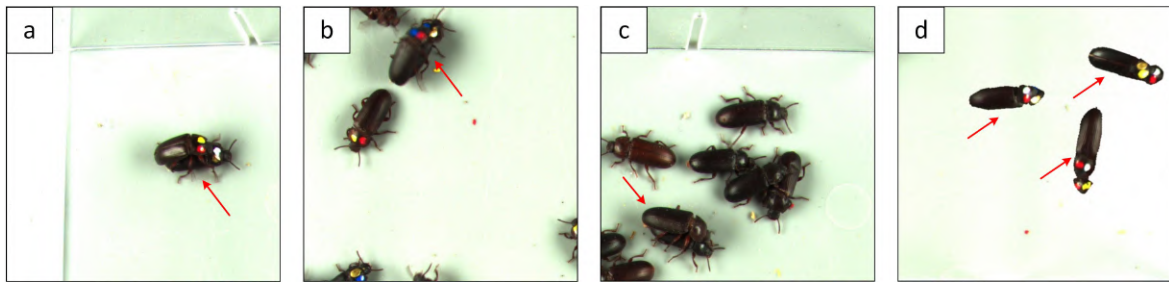


simultaneously evaluated beetles. The study tested various combinations of the parameters  $\alpha$ ,  $\beta$ ,  $m_{ReID}^{thresh}$ ,  $SSIM_{thresh}$  and  $a$ . Details about the evaluation of beetle initial selection based on ranking are described in the section 2.15.

### 2.13. Detection of beetle mating

One element of the proposed phenotyping method is the detection of mating. The detection of this behaviour pattern during the test stage can be used to determine the sex of individuals further and to select the individuals most willing to reproduce from the population.

In the study, the mating pattern was detected in the recorded images using a YOLOv8 model adapted for instance segmentation. Five types of YOLOv8 models were evaluated in the study: n, s, m, l and x, which varied in complexity. Manual labeling was performed for all recorded images of beetles with and without tags. A total of 173 mating patterns were labelled on the different views, resulting in 443 tiles containing the mating pattern and 2518 tiles not containing the mating pattern. The developed set of images provided the basis for determining the training, validation and test subsets for successive splits. Details of the evaluation of the mating detection models are presented in section 2.15. Examples of mating patterns from the developed image set are shown in Figure 10. Figure 10 shows the 640x640 tiles that were the input to the instance segmentation models for the beetle mating pattern detection.



**Figure 10:** Examples of mating patterns from the developed dataset: (a)-(c) real images, and (d) generated synthetic image.

Additionally, a method was developed to generate synthetic data to increase the number of samples in the training set for the mating detection problem. The extracted images of two beetles were combined with a specific overlap along a determined orientation, which allowed a simple simulation of the mating pattern. An example of the generated synthetic image is shown in Figure 10d.

The number of epochs for training YOLOv8 models was set to 20. After the selected (20) epochs, no improvement in performance was observed on the validation set. The batch size parameter was chosen considering the capabilities of the GPU used: 2, 4 or 8, depending on the complexity of the models. Before training, oversampling was performed for samples containing a mating pattern to balance the number of tiles with a mating pattern and those without a mating pattern. The training set was assumed to be balanced regarding the number of samples related to the real images and the generated synthetic images.

### 2.14. Reduction of the impact of domain shift effect on re-identification

Between the training stage and the test stage, a problem with domain shift was noted, resulting in a noticeable decrease in re-identification performance when comparing the results obtained on the validation set (acquired in the training stage) and the results on the test set (acquired in the test stage). Domain shift phenomenon was particularly significant for beetles, which had low mobility in the training stage, resulting in a low variety of samples in the training set. This problem was also highlighted in the section 2.12.

The study proposed a method of adaptation of the re-identification model to the new data using a pseudo-labeling mechanism. After performing the entire experiment cycle for

phenotyping, a set of labelled images from the training stage and a set of unlabelled images from the test stage were obtained. The first re-identification model was obtained after training the model on the set of labelled images from the training stage (with true labels). The first model was then used to make a prediction on the beetle images from the test stage. The prediction included using a k-nearest neighbors algorithm based on a feature space complemented by samples from the training stage, where the number of features depended on the type of model architecture used. Each prediction's probability  $p$  was also estimated as the ratio of the number of samples of the predicted (majority) class among the  $k$  nearest neighbors to the number of  $k$  nearest neighbors. Samples from the test stage were then selected for which the probability of prediction was greater than the threshold probability value  $p_{thresh}$ , i.e.  $p > p_{thresh}$ . The selected samples were used in the training of the second re-identification model. In the second re-identification model training, samples from the training stage (with true labels) and the test stage (with pseudo-labels with the appropriate probability value) were used.

The proposed adaptation method was evaluated for various combinations of parameters  $k$  and  $p_{thresh}$ . The details of evaluating the adaptation method are described in the section 2.15.

## 2.15. Evaluation

For all the evaluation procedures described, the reported results were those obtained on the test set. In the case of repetition (cross-validation), the results were refereed in the form of averaged metrics values given with standard deviation.

### 2.15.1. Segmentation of the head and abdomen of beetles

For the task of segmenting the head and abdomen of beetles, 4 splits were proposed for the training set and the test set, with the test set consisting of samples from a different acquisition series in each iteration. From the training set, 30% of samples were extracted for the validation set. Independence between the training and validation collections was ensured at the level of the individual frames from which the tiles were extracted. Standard metrics for object detection tasks were used, i.e., average precision at IoU 50% (AP50), mean average precision (mAP50:95) and F1-score (for optimal working point). The best training epoch was selected based on the results on the validation set. The training and evaluation procedure was repeated for each YOLOv8 architecture. Details of the calculated metrics can be found in Padilla et al. (2020).

### 2.15.2. Classification of detected abdomens into occluded and non-occluded

Quantitative evaluation of the abdomen classification model into occluded/non-occluded was carried out using cross-validation with the number of splits equal to 3 - in each iteration a different part of the dataset was used as a test set containing about 1/3 of the samples. From the training set, 30% of samples were extracted for the validation set. Two threshold-independent metrics were used to evaluate the classification model: the area under the precision-recall curve (AUCPR) and the area under the ROC curve (AUCROC). The ROC curve (receiver operating characteristic curve) determines the relationship between the true positive rate (TPR) and the false positive rate (FPR) for different threshold values. Additionally, the F1-score metric (related to the optimal working point) was calculated. The best training epoch was selected based on the results on the validation set using the AUCPR metric. The training and evaluation procedure was repeated for each backbone.

### 2.15.3. Determination of the orientation of the beetle

To quantitatively evaluate orientation determination, a set of manually labelled beetles (with the orientation of the major axis marked) was split into a training-validation set (70% of samples) and a test set (30% of samples). Using the training-validation set, cross-validation was performed at 5 splits for fine-tuning the  $\alpha_{T1}$  and  $\alpha_{T2}$  parameters. Mean absolute error (MAE) was used as a metric, defining the mean absolute difference between the true orientation (expressed by the angle in degrees) and the orientation determined by the proposed method. For the fine-tuned  $\alpha_{T1}$

and  $\alpha_{T_2}$  values, a final evaluation of the orientation determination method was performed on the test set.

#### 2.15.4. Automatic tag recognition

Two quantitative evaluations were carried out for the automatic tag recognition problem. The first evaluation was related to selecting the best combination (machine learning model, color space) for the classification of pixels (semantic segmentation) belonging to the beetle head. This experiment used stratified cross-validation at 5 splits with F1-macro (averaged F1 for each class) metric. Splits were determined at the level of the annotated areas described in Table 2. Results were reported as averaged F1-macro values over the splits with the standard deviation. The best combination of settings obtained was used to evaluate the automatic tag reading method overall. The second (overall) evaluation compared the automatically determined tag by the proposed method and the tag read by the user. The percentage of correctly determined tags was referred to as the results of this evaluation.

#### 2.15.5. Re-identification

In the case of re-identification, the division into training/validation and test sets was related to the successive stages of phenotyping: training and test, which ensured the complete independence of the data in these sets. To carry out repetitions for training the re-identification model and introduce variance in the training set, the samples collected in the training stage were divided into 5 approximately equal parts associated with a specific acquisition time interval. In each repetition, a different part functioned as the validation set. For the quantitative evaluation of re-identification models, 4 metrics were used: (1) mean average precision (MAP), (2) mean average precision at R (MAP@R), (3) precision at 1 (P@1), and (4) R-precision (RP). The P@1 metric was used for qualitative evaluation, as it was the easiest to interpret (the probability that the determined ID based on the nearest neighbor is correct). The rest of the three metrics also took into account (unlike P@1) reference samples that were not directly used during the inference (other neighbors), which more broadly describes the model's (embeddings) performance and provides metrics robustness to local noise. The version of metrics with the addition of "at R" assumes that only a selected number  $R$  of reference samples is used to calculate the metric's value, the validity of which has been argued by researchers (Musgrave et al. (2020a)). The best training epoch of the re-identification model was selected based on the results on the validation set using the MAP@R metric. The referenced metrics are averaged values over the values obtained for consecutive individuals, with the weight for each individual in calculating the average being the same. An implementation of metrics from the metric-learning library (Musgrave et al. (2020b)) was used. In the documentation of the metric-learning library, further details can be found regarding determining the values of specific metrics.

#### 2.15.6. Initial selection of beetles for re-identification

The basis for quantitative evaluation of the method for the initial selection of beetles for re-identification (under different settings) was the characteristic  $P@1(n_{sim})$  - the change of P@1 depending on the number of simultaneously analyzed beetles  $n_{sim}$ . The number of characteristic  $P@1(n_{sim})$  points corresponded to the number of individuals, i.e. 80. Based on the characteristic, the area under the curve (AUCPN) and the values of  $n_{sim}$  were calculated for a certain threshold of the average value of  $P@1$ , i.e. for the thresholds considered in the study of 0.85, 0.90, 0.95, the parameters  $n_{sim}^{0.85}$ ,  $n_{sim}^{0.90}$ , and  $n_{sim}^{0.95}$  were determined, respectively. The parameters  $n_{sim}^{0.85}$ ,  $n_{sim}^{0.90}$ , and  $n_{sim}^{0.95}$  were calculated by averaging the values of  $n_{sim}$  for the 5 characteristic points closest to the threshold under consideration. The area under the curve (AUCPN) was calculated by averaging the  $P@1$  values over all  $n_{sim}$ . To determine  $P@1(n_{sim})$  characteristic points, 5 re-identification models were used from cross-validation at the settings for which the best results for re-identification were achieved. Each  $P@1(n_{sim})$  characteristic point represented results averaged over splits (different models). The rankings (in the standard version of the initial

selection) were determined based on the results on the validation set (part of the images acquired in the training stage). For the baselines considered, the rankings were determined randomly for the lower baseline and based on the results on the test set for the upper baseline. The  $P@1(n_{sim})$  characteristic was smoothed with a moving average using a sliding window of size 3.

#### 2.15.7. *Detection of beetle mating*

Quantitative evaluation of beetle mating detection was carried out for specific YOLO models with the setting 'use synthetic', determining whether synthetic data were added to the training set. Independence between the training-validation and test sets was ensured at the level of the different acquisition series. Three test sets were defined using the best-represented sets (with the largest number of samples) from the three selected series. Independence between the training and validation collections was ensured at the level of the individual frames from which the tiles were extracted. A total of 9 separate model trainings were carried out for a specific combination (YOLO model type, parameter 'use synthetic'). For each of the three test sets, training was repeated 3 times for a different split between the training and validation sets. As metrics for this experiment, average precision at IoU 50% (AP50), mean average precision (mAP50:95) and F1-score (for the optimal working point) were used. Metrics were calculated using the determined bounding boxes (Box).

#### 2.15.8. *Processing time*

For training and inference of the developed model, hardware with the following parameters was used: GeForce RTX 2060 SUPER 8GB (GPU) and AMD Ryzen 7 1700 3GHz (CPU). Processing time analyses were based on times assuming batch mode processing of single images at the largest possible batch size. When referencing inference time, only steps relevant to the final solution (phenotyping beetles without physical tags) were included, i.e. (1) abdomen/head segmentation, (2) abdomen occluded/non-occluded classification, (3) orientation determination, (4) abdomen/head merging, (5) re-identification, and (6) mating detection. Total inference times were referenced as unit time (per tile or per beetle) and as total time (per frame) for phenotyping beetles in the test stage. In the analysis of processing time, simultaneous analysis of 20 beetles was assumed (as in the acquisition series), and the number of tiles analyzed was 63, which made it possible to analyze the whole recorded images using 25% overlaps between adjacent tiles in order to eliminate boundary effects. In the case of re-identification, processing time included extracting embeddings and finding  $k$ -nearest neighbors among gallery samples at  $k = 10$ .

### 3. Results and Discussion

This section presents the quantitative results for the following issues addressed in the following order: (1) detection and segmentation of the head and abdomen of beetles, (2) classification of detected abdomens into occluded and non-occluded, (3) determination of the orientation of the beetle, (4) automatic tag recognition, (5) development of a model for the re-identification of beetles for the test stage, (6) ablation studies for re-identification, (7) initial selection of beetles for re-identification for the test stage, (8) detection of beetle mating, (9) reduction of the impact of domain shift effect on re-identification, (10) further possibilities to expand the method for beetles phenotyping, and (11) processing time.

#### 3.1. *Detection and segmentation of the head and abdomen of beetles*

Quantitative results for the problem of detection and segmentation of the head and abdomen of beetles are presented in Table 4.

Based on the results in Table 4, it can be seen that the best results were achieved for the YOLOv8s-seg model while using the smaller YOLOv8n-seg model will not contribute to a noticeable reduction in accuracy and will result in shorter inference times. The YOLOv8n-seg model for detection and segmentation of the head and abdomen of beetles was selected for further analysis. The very high results obtained for the segmentation of the head/abdomen allow us

**Table 4**

Results for beetle head and abdomen detection.

model name	class names	AP50(Box)	mAP50:95(Box)	F1-score(Box)
YOLOv8n-seg	head/abdomen	0.990 $\pm$ 0.001	0.794 $\pm$ 0.013	0.971 $\pm$ 0.003
YOLOv8s-seg	head/abdomen	<b>0.991</b> $\pm$ 0.002	<b>0.801</b> $\pm$ 0.012	<b>0.972</b> $\pm$ 0.007
YOLOv8m-seg	head/abdomen	0.988 $\pm$ 0.003	0.793 $\pm$ 0.015	0.968 $\pm$ 0.011
YOLOv8l-seg	head/abdomen	0.989 $\pm$ 0.002	0.793 $\pm$ 0.014	0.964 $\pm$ 0.009
YOLOv8x-seg	head/abdomen	0.988 $\pm$ 0.004	0.791 $\pm$ 0.015	0.962 $\pm$ 0.009

to assume that this pipeline element will not significantly affect the performance of the final solution.

### 3.2. Classification of detected abdomens into occluded and non-occluded

Quantitative results for the problem of classification of detected abdomens into occluded and non-occluded are presented in Table 5.

**Table 5**

Results for classification of detected abdomens into occluded and non-occluded.

backbone name	class names	AUCPR	AUCROC	F1-score
ResNet18	occluded/non-occluded	<b>0.942</b> $\pm$ 0.024	<b>0.992</b> $\pm$ 0.001	<b>0.904</b> $\pm$ 0.011
ResNet50	occluded/non-occluded	0.877 $\pm$ 0.043	0.980 $\pm$ 0.007	0.837 $\pm$ 0.031
ResNet101	occluded/non-occluded	0.833 $\pm$ 0.100	0.978 $\pm$ 0.015	0.830 $\pm$ 0.064
MobileNetV2	occluded/non-occluded	0.933 $\pm$ 0.019	0.991 $\pm$ 0.001	0.878 $\pm$ 0.020
EfficientNetB0	occluded/non-occluded	0.814 $\pm$ 0.028	0.968 $\pm$ 0.006	0.756 $\pm$ 0.023
EfficientNetB4	occluded/non-occluded	0.859 $\pm$ 0.014	0.979 $\pm$ 0.005	0.833 $\pm$ 0.020

Based on the results in Table 5, it can be observed that the best results were achieved for the ResNet18 model, which was selected for further analysis. The results obtained for the present classification problem made it possible to effectively filter out non-occluded from the occluded abdomens. The errors made by the classification model mainly were related to borderline examples, e.g., the abdomen with a small occlusion. Re-identification for such examples should still make sense. However, the effect of occlusion on re-identification performance should be explored in more detail in future work.

### 3.3. Determination of the orientation of the beetle

Qualitative results for the problem of determining beetle orientation are shown in Figure 11. Figure 11 shows selected images of beetles with identified characteristic points and positions for the major and minor axes. The orientation of the major axis was used then in the normalization of the beetle rotation before re-identification and in the automatic tag recognition method.

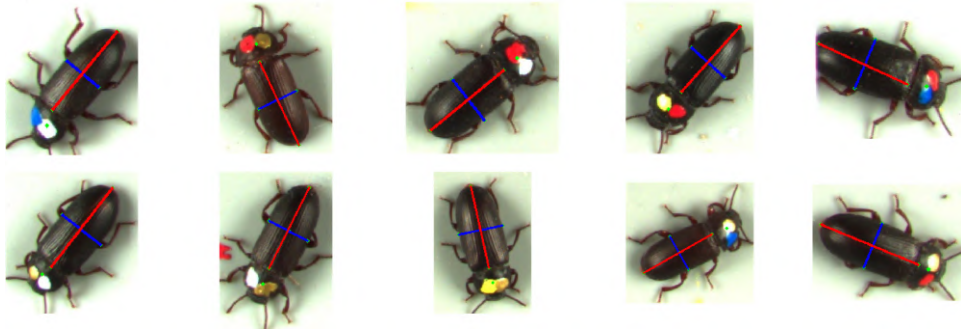
After fine-tuning, the parameter values were set to  $\alpha_{T_1} = 1^\circ$  and  $\alpha_{T_2} = 1^\circ$ . The MAE error between the manually marked and the determined orientations was  $1.76^\circ \pm 1.48^\circ$ . The results shown in Figure 11 and the low MAE values confirm the high performance of this processing step.

### 3.4. Automatic tag recognition

The results for the automatic tag recognition problem are presented in Figure 12 and in Table 11 (can be found in the Appendix). Table 11 shows the results for searching the best combination (machine learning model, color space) for a semantic segmentation task. Figure 12 shows qualitative results (successive steps of the proposed method) for selected samples (individual beetles).

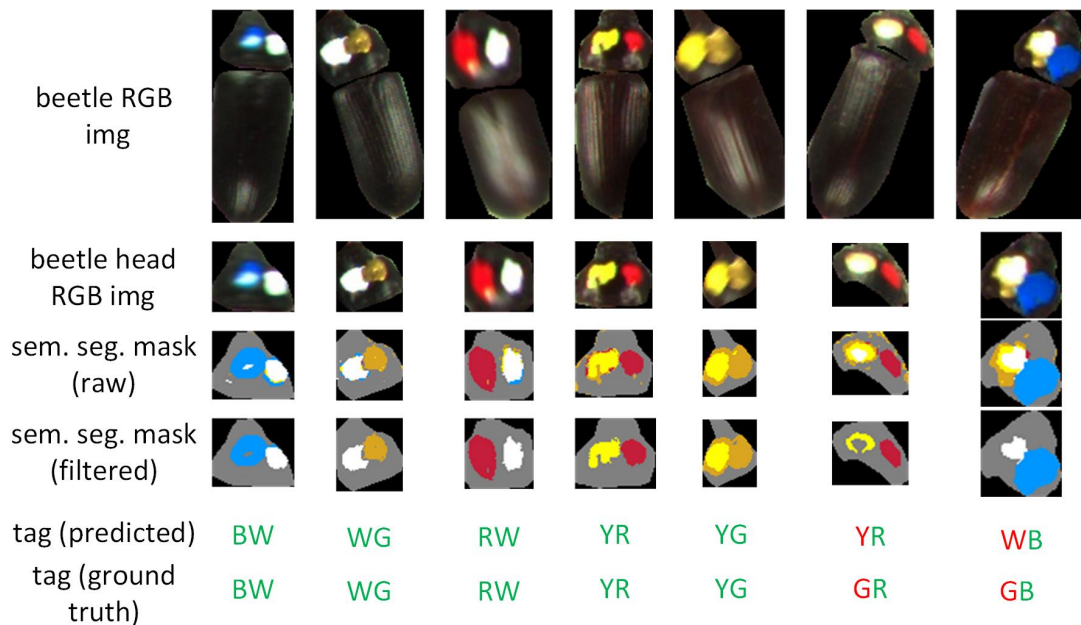


### Phenotyping of Tenebrio Molitor beetles



**Figure 11:** Results of the proposed method for determining beetle orientation by identifying characteristic points and the position of the major (red line) and minor (blue line) axes for selected samples.

Based on the results in Table 11 (in Appendix), it can be concluded that the best results for semantic segmentation were obtained using the SVM rbf model to classify pixels from images after transformation to HSV color space. These settings were used to perform semantic segmentation for selected samples from the test set. Figure 12 shows the semantic segmentation results for the selected samples from the test set (raw and after filtering). Among the 111 samples evaluated, 5 errors were registered, resulting in a 95.5% accuracy of the automatic tag recognition method. All errors were related to mistakes between white/yellow/gold colors when overexposure occurred. The overexposure problem is shown for selected samples in Figure 12. Using other tag colors (e.g. green, orange) in the future should eliminate the problem completely. The qualitative and quantitative results confirmed the proposed method's effectiveness for automatic tag recognition. The automatic tag recognition method significantly reduced the time to develop a dataset for beetle re-identification. The few errors that occurred were eliminated by manual inspection of the samples.



**Figure 12:** Results for automatic tag recognition as visualisation of the successive steps of the proposed method for selected samples (individual beetles).

### 3.5. Development of a model for the re-identification of beetles for the test stage

The quantitative results of beetle re-identification are shown in Table 6 (comparison of the proposed approach for re-identification of beetles with lower and upper baselines) and in Table 13 in Appendix (results of fine-tuning parameters for re-identification model and training).

**Table 6**

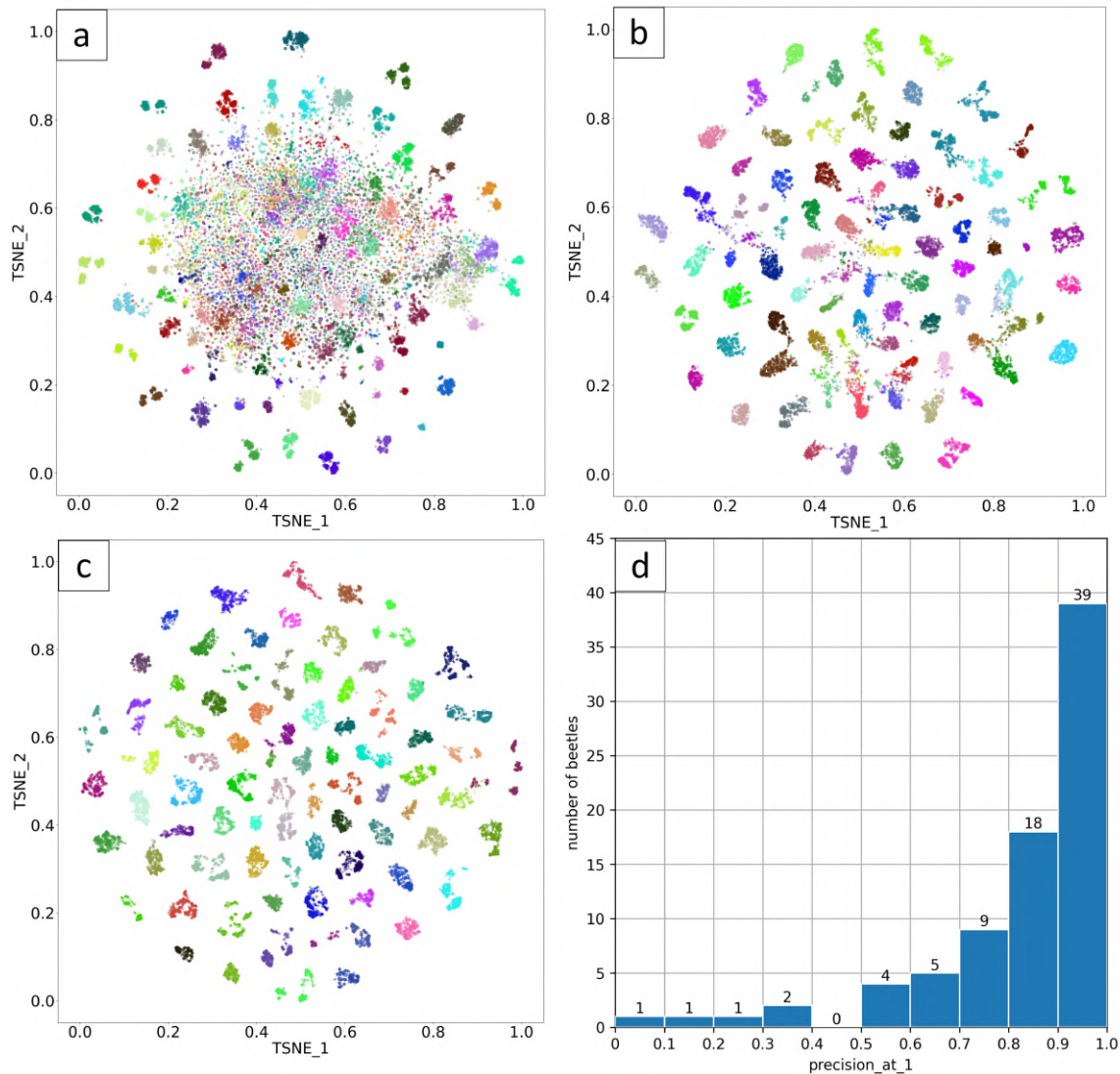
Comparison of the proposed approach for re-identification of beetles with lower and upper baselines.

approach name	visible part	fine-tuning	MAP	MAP@R	P@1	RP
standard (fine-tuned backbone)	abdomen	yes	0.818 $\pm$ 0.017	0.770 $\pm$ 0.021	0.807 $\pm$ 0.022	0.791 $\pm$ 0.019
lower baseline v1 (frozen backbone)	abdomen	no	0.128 $\pm$ 0.001	0.063 $\pm$ 0.001	0.185 $\pm$ 0.005	0.128 $\pm$ 0.002
lower baseline v2 (handcrafted features)	abdomen	-	0.043 $\pm$ 0.001	0.019 $\pm$ 0.001	0.130 $\pm$ 0.002	0.051 $\pm$ 0.001
upper baseline	abdomen+head (with tag)	yes	0.962 $\pm$ 0.002	0.947 $\pm$ 0.003	0.959 $\pm$ 0.003	0.953 $\pm$ 0.003

The fine-tuning of beetle re-identification model parameters and training (see results in Table 13 in Appendix) finally resulted in the selection of the following optimal settings: MobileNetv2 model with one hidden layer consisting of 1024 neurons trained with Triplet Margin Loss and Triplet Margin Miner. The margin values were set at 0.2 for loss and miner. The best results were achieved with cosine distance and using all triplets in training (no filtering of triplets). Based on the results in Table 6, it can be concluded that for the beetle re-identification problem being addressed, developing a problem-oriented feature extractor (fine-tuned) is necessary to obtain satisfactory results. The results obtained when using a frozen backbone (lower baseline 1) or handcrafted features (lower baseline 2) extractor were very low. As expected, relatively high results were obtained for the approach of using beetle images with tags (abdomen+head) when fine-tuning the extractor (upper baseline). The reasons for the mistakes that occurred with this approach should be sought in the repetition of tag colors in successive series, i.e. among the 80 beetles considered, each tag repeated 4 times (4 separate phenotyping series). In Figure 13a-c, the charts for TSNE analysis (Van der Maaten and Hinton (2008)) for the considered approaches are shown. In Figure 13a-c, each color is associated with a different individual, and the type of tag (circle or plus) indicates the training or test stage of the phenotyping.

The results in Figure 13a-c confirm the quantitative results in Table 6. It is worth noting the very good separability of classes in feature space in the case of chart 13b, which is related to the proposed approach when only images of the abdomen are used for re-identification. In the case of 13c, an almost perfect separation of the considered classes was achieved. Figure 13d additionally shows the re-identification precision distribution for the analyzed individuals. As many as 57/80 individuals had a re-identification precision higher than 0.8, and only 5/80 individuals were particularly difficult to re-identify (precision less than 0.5). The variation within re-identification precision for individuals was the main motivation for proposing a method for the initial selection of beetles for re-identification.

Figure 14 shows examples of re-identification model predictions, distinguishing TP (with estimated probability equal to 1 and less than 0.9), FN and FP predictions.



**Figure 13:** Results for beetle re-identification: (a) TSNE analysis for lower baseline v1 (frozen backbone and abdomen images) approach, (b) TSNE analysis for standard (fine-tuned backbone and abdomen images) approach, (c) TSNE analysis for upper baseline (fine-tuned backbone and images of whole beetle with tag) approach, and (d) histogram for re-identification metric for individual beetles under standard approach.

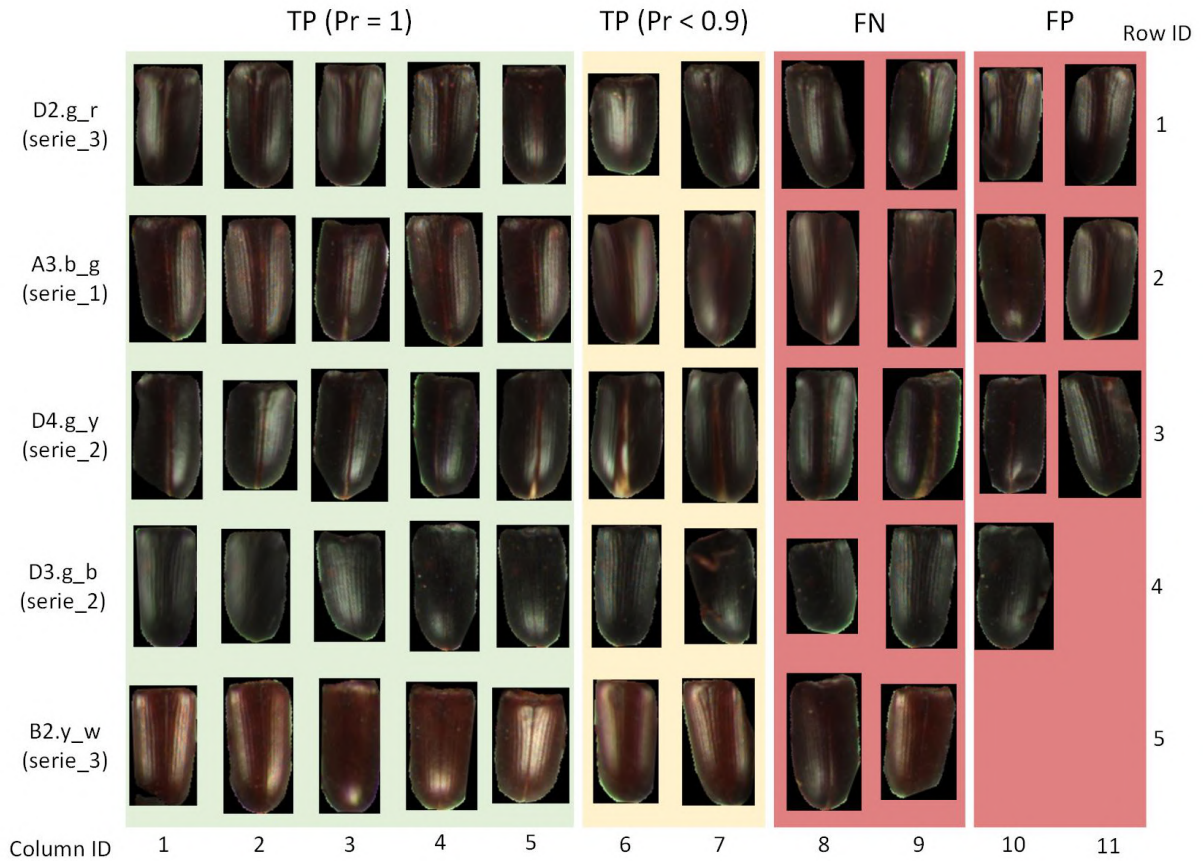
Analyzing the predictions in Figure 14, it can be observed that there is a wide variation among TP predictions, confirming the re-identification model's robustness. Among the most important reasons for mistakes (or reduction in prediction probability) by the re-identification model are (sample references are in the form [number row].[number column]): (1) significant change in pose (samples 1.6., 1.8, and 4.8), (2) segmentation errors (sample 4.10.), (3) significant change in abdomen structure (samples 3.6, and 3.9), (4) motion blur (samples 2.6, 2.7, and 5.6), (5) significant change in lighting conditions (sample 5.8), and (6) contamination (sample 4.7.).

Figure 15 presents the beetles ranked according to ease of re-identification (10 easiest and 10 most difficult to re-identify), taking into account the results on the test set.

In the case of the 10 beetles easiest to re-identify, it is important to note the characteristics in the appearance of these individuals, which undoubtedly facilitated re-identification, including color characteristics (e.g. in the case of beetles with ranks 2 and 6, we can see brown spots on the black abdomen), relatively bright abdomen (e.g., for beetles with ranks 3, 4, and 5, we observe brown abdomens), and abdomen structure characteristics (e.g., for beetle with rank 4, we



## Phenotyping of *Tenebrio Molitor* beetles



**Figure 14:** Examples of re-identification model prediction: TP (with confidence equal to 1), TP (with confidence less than 0.9), FN and FP errors.

see dent). It should be emphasized that appearance features were only one element determining the ease of re-identification. The other important element was the variation in training samples obtained during the training stage (when the beetles were isolated). For the most difficult beetles to re-identify, 8/10 were characterized by lack of (or very low mobility) during the training stage, which was the main reason for the difficulty of re-identification in this case. Future work should consider strategies to ensure that the training samples are varied for each beetle, e.g. through forced changes in the acquisition conditions, i.e. lighting and camera position.

### 3.6. Ablation studies for re-identification

Quantitative results of ablation studies for beetle re-identification are summarized in Table 7.

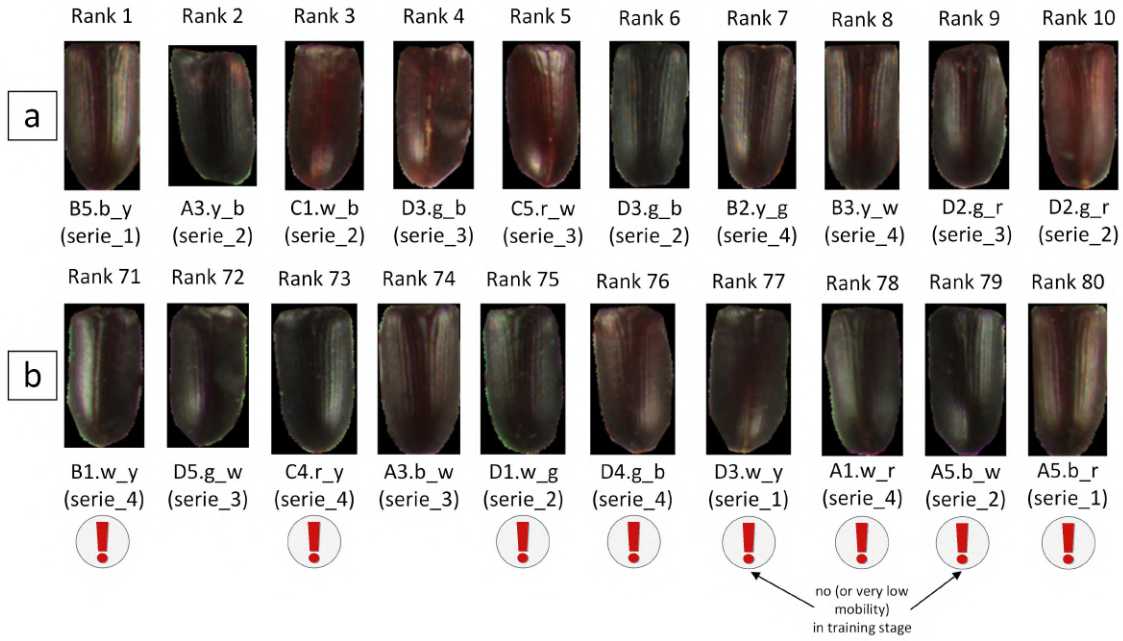
**Table 7**

Results of ablation studies for beetle re-identification.

approach name	MAP	MAP@R	P@1	RP
S	0.083 $\pm$ 0.003	0.032 $\pm$ 0.003	0.060 $\pm$ 0.003	0.058 $\pm$ 0.002
S+T+I	0.367 $\pm$ 0.030	0.301 $\pm$ 0.031	0.339 $\pm$ 0.026	0.323 $\pm$ 0.031
S+I	0.321 $\pm$ 0.025	0.254 $\pm$ 0.024	0.292 $\pm$ 0.025	0.279 $\pm$ 0.025
S+I+C	0.781 $\pm$ 0.009	0.728 $\pm$ 0.012	0.763 $\pm$ 0.010	0.750 $\pm$ 0.011
S+T+I+C (baseline)	0.818 $\pm$ 0.017	0.770 $\pm$ 0.021	0.807 $\pm$ 0.022	0.791 $\pm$ 0.019

The results in Table 7 clearly show that color-related features had the greatest impact on beetle re-identification, as evidenced by the significant difference between the results obtained for the S+T+I (grayscale images) and S+T+I+C (raw RGB images) approaches - MAP=0.367

## Phenotyping of *Tenebrio Molitor* beetles



**Figure 15:** Results for beetle re-identification: (a) 10 beetles easiest to re-identify (1-10 positions in ranking), (b) 10 beetles most difficult to re-identify (71-80 positions in ranking).

for S+T+I and MAP=0.818 for S+T+I+C. By comparing the results for the S+I/S+T+I and S+I+C/S+T+I+C approaches, it is possible to assess the influence of texture features on the re-identification of beetles. In both cases, it can be observed that texture features had a positive effect on re-identification results. However, still, the texture had a much smaller impact than color features, i.e., considering texture for re-identification contributed to an increase in MAP from 0.321 to 0.367 for S+I/S+T+I and from 0.781 to 0.818 for S+I+C/S+T+I+C. Basing only on shape features (S approach) did not allow the re-identification of beetles. At this point, it is worth mentioning that in the paper (Murali et al. (2019)) very high results were achieved for the re-identification of fruit flies when based on grayscale images, while it was not shown in the paper what mainly affected the ability to re-identify insects. Undoubtedly, the results shown for beetle re-identification confirm the importance of conducting ablation studies for insect re-identification problems, which enables a deeper understanding of how the re-identification model works.

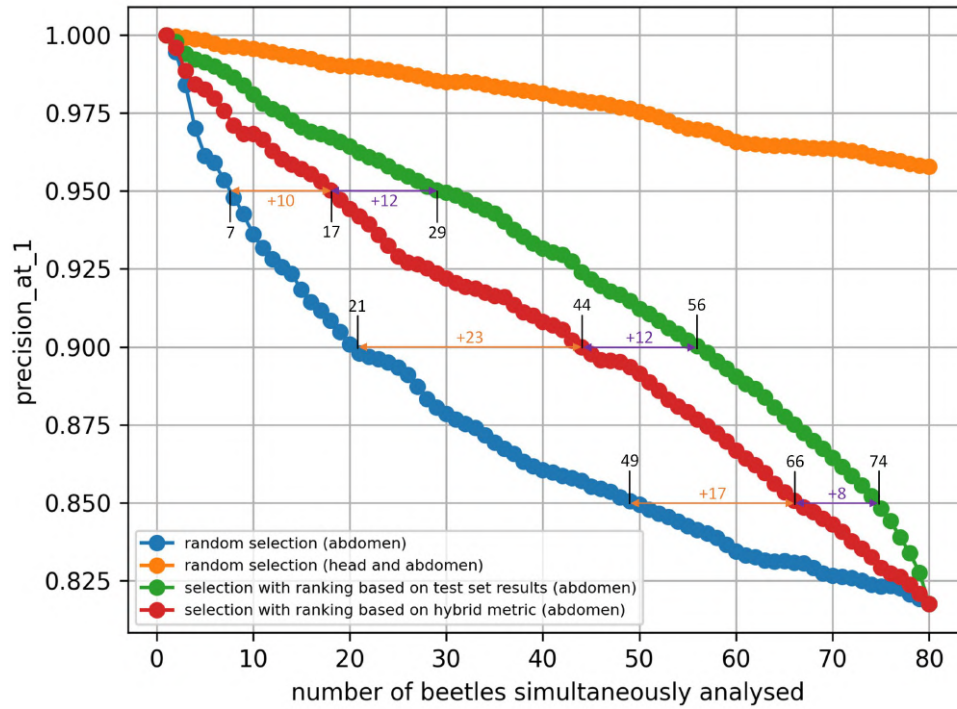
### 3.7. Initial selection of beetles for re-identification for the test stage

The results for the proposed beetle initial selection strategy for re-identification are included in Table 14 (in Appendix) and Figure 16. Table 14 contains quantitative details from subsequent parameter fine-tuning steps for the approach based on the proposed hybrid metric. Figure 16 summarizes the results obtained, comparing the proposed approach (red curve) with the lower (random selection, blue curve) and upper baseline (selection based on results on the test set, green curve). Figure 16 also shows results for re-identifying beetles when considering the use of physical tags (orange curve).

After parameter fine-tuning for the hybrid metric approach for the initial selection (see Table 14), the following optimal parameters were obtained:  $m_{ReID}^{thresh} = 0.7$ ,  $SSIM_{thresh} = 0.8$ ,  $a = 40$ ,  $\alpha = 1$  and  $\beta = 4$ . Figure 16 confirms that the use of the initial selection of beetles for re-identification can allow a significant increase in the number of simultaneously analyzed beetles at a given level of average precision for re-identification. For example, for a threshold of average precision set at 0.9, the proposed approach would allow increasing the number of simultaneously analyzed beetles from 21 to 44, comparing the proposed approach with the lower



## Phenotyping of Tenebrio Molitor beetles



**Figure 16:** Chart showing the change in re-identification precision as a function of the number of beetles analyzed simultaneously in the phenotyping procedure with different approaches to the initial selection of beetles.

baseline (random selection). It should also be noted that the obtained number of beetles does not deviate significantly from the upper baseline (selection based on results on the test set), i.e. the difference is only 12 additional individuals. The obtained results for the proposed initial selection approach confirm the validity of introducing this step into the phenotyping procedure, especially since the pool of possible objects (available beetles) for phenotyping will not be a limitation in most cases.

### 3.8. Detection of beetle mating

The model evaluation results for the beetle mating detection problem are summarized in Table 8.

**Table 8**

Results for detection of beetle mating.

model name	class names	use synthetic	AP50(Box)	mAP50:95(Box)	F1-score(Box)
YOLOv8n-seg	mating	0	0.693 ± 0.092	0.567 ± 0.097	0.669 ± 0.063
YOLOv8n-seg	mating	1	0.815 ± 0.035	0.690 ± 0.041	0.756 ± 0.041
YOLOv8s-seg	mating	0	0.718 ± 0.101	0.607 ± 0.097	0.687 ± 0.087
YOLOv8s-seg	mating	1	0.778 ± 0.055	0.670 ± 0.057	0.731 ± 0.031
YOLOv8m-seg	mating	0	0.754 ± 0.060	0.633 ± 0.060	0.729 ± 0.034
YOLOv8m-seg	mating	1	0.794 ± 0.070	0.684 ± 0.057	0.760 ± 0.045
YOLOv8l-seg	mating	0	0.774 ± 0.149	0.656 ± 0.142	0.750 ± 0.134
YOLOv8l-seg	mating	1	<b>0.835 ± 0.067</b>	<b>0.732 ± 0.049</b>	<b>0.777 ± 0.063</b>
YOLOv8x-seg	mating	0	0.684 ± 0.127	0.578 ± 0.106	0.676 ± 0.081
YOLOv8x-seg	mating	1	0.753 ± 0.110	0.660 ± 0.093	0.706 ± 0.112

For the beetle mating detection problem, the best results were obtained for the YOLOv8l-seg model when applying the proposed augmentation technique (using generated synthetic images

to train the model). It is also worth noting that the addition of synthetic images to the training set for all models enabled improved detection accuracy. The results confirmed the ability to detect the mating pattern on single shots. Future work should focus on expanding the dataset of beetle mating examples. Due to the relatively high performance of the detection model obtained in this work, the labeling process can be effectively speeded up using a weak model when assessing the relevance of a large amount of unlabeled data. It is also worth considering other methods for mating detection, e.g. based on temporal data and supported by tracking. Undoubtedly, the results of the detection of the mating pattern of beetles obtained in this study confirmed the validity of including this element in the procedure of beetles phenotyping.

### 3.9. Reduction of the impact of domain shift effect on re-identification

Quantitative results for the proposed method of reducing the effect of domain shift on re-identification are summarized in Table 15 and Table 9. Table 15 in the Appendix shows the results of parameter fine-tuning for the proposed method, and Table 9 summarizes the results for domain shift problem, comparing the results with (for the best parameter combination) and without using the domain adaptation mechanism.

**Table 9**

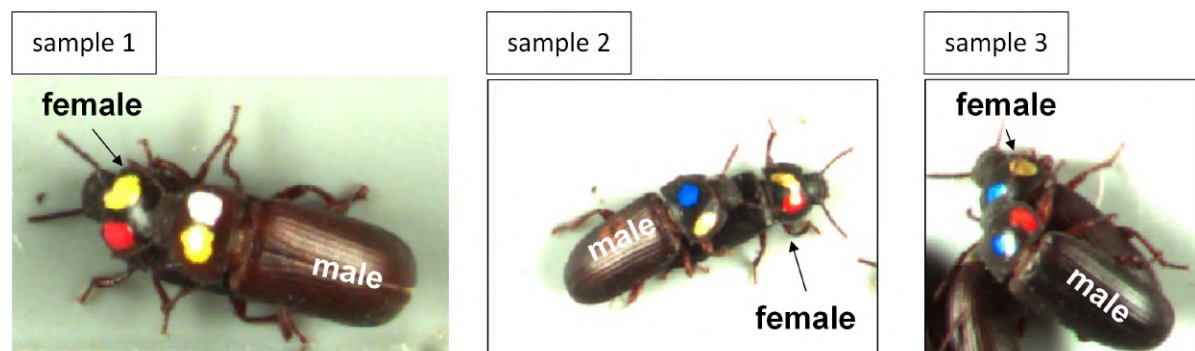
The impact of the proposed domain adaptation method on beetle re-identification results.

use adaptation	MAP	MAP@R	P@1	RP
no	0.818 $\pm$ 0.017	0.770 $\pm$ 0.021	0.807 $\pm$ 0.022	0.791 $\pm$ 0.019
yes	0.866 $\pm$ 0.018	0.838 $\pm$ 0.022	0.853 $\pm$ 0.019	0.847 $\pm$ 0.020

The results in Table 9 show that the proposed method significantly improved the re-identification results by adapting the model to the data acquired during the test stage (MAP increase from 0.818 to 0.866). The best adaptation results were achieved with parameters  $k = 100$  and  $p_{thresh} = 0.95$ .

### 3.10. Further possibilities to expand the method for dynamic phenotyping of beetles

The very promising results for the beetle re-identification and mating detection task make it possible to consider applying the techniques proposed in this article to another problem that is very important from the breeder's point of view, namely determining the sex of beetles. The idea of the solution for determining the sex of beetles is presented in Figure 17.



**Figure 17:** The idea of sex determination based on detected mating patterns.

The idea presented in Figure 17 assumes that when mating is detected, the two beetles' relative position (up/down) is determined. In the case of a male individual during mating, the abdomen is usually visible in full, allowing re-identification to be carried out. In the case of a female individual, sex determination could be carried out after mating, e.g., supported by

tracking techniques. Validation of the proposed method should be based on the labels established by the specialist in manual sex verification. Undoubtedly, adding a part related to the sex determination of beetles to the phenotyping procedure is an important direction for future work.

### 3.11. Processing time

The results regarding the analysis of processing time for each processing step are shown in Table 10.

**Table 10**

Processing time for each image processing step.

ID	processing step	model type	unit time (per tile or per beetle)	total time (per frame)
1	abdomen/head segmentation	YOLOv8n-seg	4ms	0.24 s
2	abdomen occlusion classification	ResNet18	< 1ms	< 0.01 s
3	orientation determination	-	52 ms	1.04 s
4	abdomen/head merging	-	-	0.56 s
5	re-identification	MobileNetV2	1 ms	0.02 s
6	mating detection	YOLOv8l-seg	28 ms	1.75 s
summary				3.62 s

Based on the results in Table 10, it is possible to locate the bottlenecks in the whole system, considering the processing time, which was mating detection and orientation determination. For the mating detector, a relatively large model based on the YOLOv8l-seg architecture was used, as the best detection results were achieved for this model. In the future, with the expansion of the dataset with new examples of mating, the accuracy of smaller models is expected to increase, making it reasonable to replace the current model with another one with lower complexity. For orientation determination, the approach based on classical image processing can be replaced in the future by a small regression convolutional network trained on the output of the current orientation determination method, which should significantly reduce the processing time for this step. It is also worth noting that the key processing element related to re-identification has very short processing times and can even be used in near real-time processing.

## 4. Conclusions

The study proposed phenotyping method with dynamic characteristics determination for *Tenebrio Molitor* beetles in selective breeding studies based on re-identification and computer vision. The study showed that re-identification of *Tenebrio Molitor* beetles based on images of the abdomen only (without tags) is possible with satisfactory accuracy using automated image acquisition for training the re-identification model (proposing the training stage when the beetles were isolated). Physical tags enabled reliable experiments and validation by easily identifying individuals during the phenotyping test stage. The proposed methods of initial selection of beetles and reduction of the domain shift effect made it possible to enhance re-identification performance further. The ablation studies and analyses of sample predictions showed the high importance of color features for re-identification. Also, they excluded the potential basing of the re-identification model on undesirable features. The studies also showed the key role of training data quality on the accuracy of the re-identification model, which was exploited by proposing an initial beetle selection strategy. Promising results for behavioural pattern detection (i.e., mating) allow for consideration of further elements for the phenotyping procedure, e.g., sex determination of individuals. As the most important directions for future work, we see (1) increasing the diversity of training samples by interfering with image acquisition conditions (e.g., forced change of illumination, camera position), i.e. preventing the phenomenon of domain

shift at the level of the vision system, (2) developing a procedure for determining the sex of beetles with validation based on the labels proposed by the specialist, (3) supporting the re-identification of beetles with tracking techniques, (4) analyzing re-identification when occlusion occurs and proposing methods robust to this phenomenon, and (5) determining the impact of individual dynamic characteristics on the reproductive value of beetles (e.g., taking into account the fecundity of beetles).

#### **CRedit authorship contribution statement**

**Paweł Majewski:** Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualisation, Writing–original draft preparation, Funding acquisition. **Piotr Lampa:** Methodology, Visualisation, Writing–review and editing. **Robert Burduk:** Supervision, Writing–review and editing. **Jacek Reiner:** Supervision, Writing–review and editing. **Ta-Te Lin:** Supervision, Writing–review and editing, Project administration.

#### **Data availability**

The data used in this research is available from the corresponding author on reasonable request.

#### **Declaration of Competing Interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### **Acknowledgements**

We wish to thank Paweł Górzyński and Dawid Biedrzycki from *Tenebria* (Lubawa, Poland) for providing experimental material in the form of *Tenebrio Molitor* beetles and valuable suggestions. The work presented in this publication was carried out within the project "Non-invasive method for assessing the breeding and reproductive value of insects using re-identification and machine learning techniques", financed by a scholarship from the Visegrad-Taiwan Scholarships programme led by the International Visegrad Fund.

## A. Appendix

Table 11

Results for searching for the best combination (machine learning model, color space) for a semantic segmentation task for automatic tag recognition task.

ID	color space	model name	F1-macro
1	HSV	SVM rbf	<b>0.870</b> $\pm$ 0.019
2	YCrCb	SVM rbf	0.862 $\pm$ 0.023
3	RGB	SVM rbf	0.859 $\pm$ 0.023
4	Lab	SVM rbf	0.857 $\pm$ 0.021
5	YCrCb	SVM linear	0.851 $\pm$ 0.026
6	YCrCb	LogReg	0.849 $\pm$ 0.028
7	RGB	SVM linear	0.847 $\pm$ 0.028
8	Luv	SVM rbf	0.845 $\pm$ 0.024
9	Lab	SVM linear	0.845 $\pm$ 0.026
10	RGB	LogReg	0.840 $\pm$ 0.029
11	Lab	LogReg	0.839 $\pm$ 0.026
12	Luv	LogReg	0.838 $\pm$ 0.03
13	Luv	SVM linear	0.838 $\pm$ 0.03
14	YCrCb	LDA	0.743 $\pm$ 0.021

Table 12

Description of the stages of selection of optimal parameters for the beetle re-identification model.

ID	backbone	FC layers	loss type	miner type	margin val in loss	margin val in miner	distance type	triplets type
1.1.	ResNet18	[512]	Triplet Margin	Triplet Margin	0.2	0.2	cosine	semihard
1.2.	ResNet50	[512]	Triplet Margin	Triplet Margin	0.2	0.2	cosine	semihard
1.3.	ResNet101	[512]	Triplet Margin	Triplet Margin	0.2	0.2	cosine	semihard
1.4.	EfficientNet-b0	[512]	Triplet Margin	Triplet Margin	0.2	0.2	cosine	semihard
1.5.	EfficientNet-b4	[512]	Triplet Margin	Triplet Margin	0.2	0.2	cosine	semihard
1.6.	MobileNetv2	[512]	Triplet Margin	Triplet Margin	0.2	0.2	cosine	semihard
2.1.	MobileNetv2	[]	Triplet Margin	Triplet Margin	0.2	0.2	cosine	semihard
2.2.	MobileNetv2	[1024]	Triplet Margin	Triplet Margin	0.2	0.2	cosine	semihard
2.3.	MobileNetv2	[512, 256]	Triplet Margin	Triplet Margin	0.2	0.2	cosine	semihard
2.4.	MobileNetv2	[1024, 512]	Triplet Margin	Triplet Margin	0.2	0.2	cosine	semihard
2.5.	MobileNetv2	[512, 256, 128]	Triplet Margin	Triplet Margin	0.2	0.2	cosine	semihard
2.6.	MobileNetv2	[1024, 512, 256]	Triplet Margin	Triplet Margin	0.2	0.2	cosine	semihard
3.1.	MobileNetv2	[1024]	Circle	Triplet Margin	-	0.2	cosine	semihard
3.2.	MobileNetv2	[1024]	GLS	Triplet Margin	-	0.2	cosine	semihard
3.3.	MobileNetv2	[1024]	Multi-Similarity	Triplet Margin	-	0.2	cosine	semihard
3.4.	MobileNetv2	[1024]	Proxy-NCA	Triplet Margin	-	0.2	cosine	semihard
3.5.	MobileNetv2	[1024]	FastAP	Triplet Margin	-	0.2	cosine	semihard
4.1.	MobileNetv2	[1024]	Triplet Margin	Triplet Margin	0.05	0.2	Euclidean	all
4.2.	MobileNetv2	[1024]	Triplet Margin	Triplet Margin	0.05	0.2	Euclidean	semihard
4.3.	MobileNetv2	[1024]	Triplet Margin	Triplet Margin	0.05	0.2	cosine	all
4.4.	MobileNetv2	[1024]	Triplet Margin	Triplet Margin	0.05	0.2	cosine	semihard
4.5.	MobileNetv2	[1024]	Triplet Margin	Triplet Margin	0.2	0.2	Euclidean	all
4.6.	MobileNetv2	[1024]	Triplet Margin	Triplet Margin	0.2	0.2	Euclidean	semihard
4.7.	MobileNetv2	[1024]	Triplet Margin	Triplet Margin	0.2	0.2	cosine	all



**Table 13**

Results for the subsequent stages of selection of optimal parameters for the beetle re-identification model.

ID	MAP	MAP@R	P@1	RP
1.1.	0.765 $\pm$ 0.009	0.717 $\pm$ 0.012	0.744 $\pm$ 0.008	0.732 $\pm$ 0.010
1.2.	0.783 $\pm$ 0.011	0.733 $\pm$ 0.012	0.765 $\pm$ 0.009	0.752 $\pm$ 0.012
1.3.	0.772 $\pm$ 0.014	0.722 $\pm$ 0.016	0.757 $\pm$ 0.015	0.742 $\pm$ 0.016
1.4.	0.655 $\pm$ 0.054	0.593 $\pm$ 0.056	0.639 $\pm$ 0.056	0.617 $\pm$ 0.056
1.5.	0.779 $\pm$ 0.055	0.724 $\pm$ 0.064	0.769 $\pm$ 0.053	0.748 $\pm$ 0.060
1.6.	<b>0.800</b> $\pm$ 0.035	<b>0.750</b> $\pm$ 0.039	<b>0.783</b> $\pm$ 0.036	<b>0.770</b> $\pm$ 0.038
2.1.	0.800 $\pm$ 0.028	0.754 $\pm$ 0.033	0.775 $\pm$ 0.031	0.764 $\pm$ 0.031
2.2.	<b>0.814</b> $\pm$ 0.018	<b>0.765</b> $\pm$ 0.020	<b>0.797</b> $\pm$ 0.017	<b>0.784</b> $\pm$ 0.019
2.3.	0.785 $\pm$ 0.019	0.732 $\pm$ 0.024	0.765 $\pm$ 0.021	0.752 $\pm$ 0.021
2.4.	0.777 $\pm$ 0.027	0.723 $\pm$ 0.028	0.759 $\pm$ 0.027	0.743 $\pm$ 0.029
2.5.	0.604 $\pm$ 0.063	0.521 $\pm$ 0.070	0.577 $\pm$ 0.071	0.553 $\pm$ 0.066
2.6.	0.711 $\pm$ 0.055	0.646 $\pm$ 0.065	0.685 $\pm$ 0.061	0.669 $\pm$ 0.060
3.1.	0.799 $\pm$ 0.019	0.749 $\pm$ 0.023	0.788 $\pm$ 0.022	0.770 $\pm$ 0.021
3.2.	0.773 $\pm$ 0.020	0.717 $\pm$ 0.023	0.750 $\pm$ 0.025	0.737 $\pm$ 0.022
3.3.	0.804 $\pm$ 0.012	0.757 $\pm$ 0.015	<b>0.792</b> $\pm$ 0.009	0.776 $\pm$ 0.013
3.4.	0.685 $\pm$ 0.060	0.653 $\pm$ 0.062	0.658 $\pm$ 0.063	0.655 $\pm$ 0.062
3.5.	<b>0.806</b> $\pm$ 0.023	<b>0.762</b> $\pm$ 0.025	0.787 $\pm$ 0.026	<b>0.778</b> $\pm$ 0.025
4.1.	0.761 $\pm$ 0.010	0.685 $\pm$ 0.010	0.756 $\pm$ 0.006	0.725 $\pm$ 0.010
4.2.	0.760 $\pm$ 0.009	0.685 $\pm$ 0.009	0.757 $\pm$ 0.009	0.721 $\pm$ 0.009
4.3.	0.767 $\pm$ 0.025	0.694 $\pm$ 0.032	0.756 $\pm$ 0.018	0.731 $\pm$ 0.028
4.4.	0.779 $\pm$ 0.012	0.712 $\pm$ 0.017	0.776 $\pm$ 0.008	0.745 $\pm$ 0.014
4.5.	0.787 $\pm$ 0.001	0.726 $\pm$ 0.001	0.767 $\pm$ 0.006	0.752 $\pm$ 0.001
4.6.	0.791 $\pm$ 0.015	0.733 $\pm$ 0.016	0.773 $\pm$ 0.014	0.759 $\pm$ 0.017
4.7.	<b>0.818</b> $\pm$ 0.017	<b>0.770</b> $\pm$ 0.021	<b>0.807</b> $\pm$ 0.022	<b>0.791</b> $\pm$ 0.019

**Table 14**

Results of beetle re-identification using initial selection strategy for various chosen parameters.

ID	$m_{ReID}^{thresh}$	$SSIM_{thresh}$	$a$	$\alpha$	$\beta$	AUCPN	$n_{sim}^{0.85}$	$n_{sim}^{0.90}$	$n_{sim}^{0.95}$
1.1.	0.5	0.6	10	1	1	0.8970	66.0	<b>41.0</b>	12.0
1.2.	0.6	0.6	10	1	1	0.8968	<b>67.0</b>	40.0	8.0
1.3.	0.7	0.6	10	1	1	<b>0.8988</b>	66.0	38.0	<b>15.3</b>
1.4.	0.8	0.6	10	1	1	0.8971	<b>67.0</b>	39.0	7.5
2.1.	0.7	0.4	10	1	1	0.8968	<b>67.0</b>	37.0	<b>13.5</b>
2.2.	0.7	0.5	10	1	1	0.8981	66.5	39.0	12.5
2.3.	0.7	0.7	10	1	1	0.8953	66.0	42.5	6.0
2.4.	0.7	0.8	10	1	1	<b>0.9001</b>	66.5	<b>44.0</b>	11.0
2.5.	0.7	0.9	10	1	1	0.8910	66.0	38.0	6.0
3.1.	0.7	0.8	5	1	1	0.8914	64.0	40.5	5.5
3.2.	0.7	0.8	20	1	1	0.8993	65.0	<b>42.0</b>	12.0
3.3.	0.7	0.8	40	1	1	<b>0.9024</b>	66.0	41.0	<b>17.0</b>
3.4.	0.7	0.8	80	1	1	0.9007	<b>66.5</b>	39.5	16.0
4.1.	0.7	0.8	40	1	0.25	0.8990	<b>66.0</b>	37.0	15.0
4.2.	0.7	0.8	40	1	0.5	0.8991	65.5	38.0	15.0
4.3.	0.7	0.8	40	1	2	0.9027	<b>66.0</b>	43.0	<b>17.0</b>
4.4.	0.7	0.8	40	1	4	<b>0.9040</b>	<b>66.0</b>	<b>44.5</b>	<b>17.0</b>
4.5.	0.7	0.8	40	1	6	0.8993	65.5	43.5	13.0
4.6.	0.7	0.8	40	1	8	0.9025	65.5	43.0	16.0

**Table 15**

Results of beetle re-identification using the proposed domain adaptation method for various chosen parameters.

ID	$k$	$P_{thresh}$	MAP	MAP@R	P@1	RP
1	20	0.70	0.826 $\pm$ 0.023	0.794 $\pm$ 0.026	0.808 $\pm$ 0.024	0.802 $\pm$ 0.025
2	20	0.90	0.844 $\pm$ 0.021	0.815 $\pm$ 0.021	0.829 $\pm$ 0.022	0.823 $\pm$ 0.021
3	20	0.95	0.847 $\pm$ 0.022	0.819 $\pm$ 0.024	0.833 $\pm$ 0.022	0.827 $\pm$ 0.023
4	20	1.00	0.854 $\pm$ 0.018	0.826 $\pm$ 0.020	0.840 $\pm$ 0.021	0.834 $\pm$ 0.019
5	50	0.70	0.831 $\pm$ 0.018	0.800 $\pm$ 0.019	0.813 $\pm$ 0.019	0.808 $\pm$ 0.019
6	50	0.90	0.851 $\pm$ 0.015	0.823 $\pm$ 0.017	0.837 $\pm$ 0.018	0.832 $\pm$ 0.017
7	50	0.95	0.858 $\pm$ 0.015	0.830 $\pm$ 0.016	0.846 $\pm$ 0.016	0.839 $\pm$ 0.016
8	50	0.97	0.861 $\pm$ 0.016	0.834 $\pm$ 0.019	0.848 $\pm$ 0.018	0.843 $\pm$ 0.018
9	50	1.00	0.865 $\pm$ 0.014	0.837 $\pm$ 0.015	<b>0.856<math>\pm</math> 0.015</b>	<b>0.847<math>\pm</math> 0.015</b>
10	100	0.70	0.834 $\pm$ 0.018	0.804 $\pm$ 0.019	0.818 $\pm$ 0.020	0.813 $\pm$ 0.018
11	100	0.90	0.855 $\pm$ 0.013	0.827 $\pm$ 0.016	0.844 $\pm$ 0.014	0.836 $\pm$ 0.015
12	100	0.95	<b>0.866<math>\pm</math> 0.018</b>	<b>0.838<math>\pm</math> 0.022</b>	0.853 $\pm$ 0.019	<b>0.847<math>\pm</math> 0.020</b>
13	100	0.97	0.863 $\pm$ 0.012	0.835 $\pm$ 0.014	0.853 $\pm$ 0.011	0.844 $\pm$ 0.014
14	100	0.99	0.865 $\pm$ 0.015	0.836 $\pm$ 0.019	<b>0.856<math>\pm</math> 0.017</b>	0.846 $\pm$ 0.017
15	100	1.00	0.864 $\pm$ 0.014	0.835 $\pm$ 0.016	<b>0.856<math>\pm</math> 0.015</b>	0.846 $\pm$ 0.015

## References

- Armitage, S., Siva-Jothy, M., 2005. Immune function responds to selection for cuticular colour in *tenebrio molitor*. *Heredity* 94, 650–656.
- Arzoumanian, Z., Holmberg, J., Norman, B., 2005. An astronomical pattern-matching algorithm for computer-aided identification of whale sharks rhincodon typus. *Journal of Applied Ecology* 42, 999–1011.
- Bhattacharya, A., Ameel, J., Waldbauer, G., 1970. A method for sexing living pupal and adult yellow mealworms. *Annals of the Entomological Society of America* 63, 1783–1783.
- Borlinghaus, P., Tausch, F., Rettenberger, L., 2023. A purely visual re-id approach for bumblebees (*bombus terrestris*). *Smart Agricultural Technology* 3, 100135.
- Cakir, F., He, K., Xia, X., Kulis, B., Sclaroff, S., 2019. Deep metric learning to rank, in: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1861–1870.
- Chan, J., Carrión, H., Mégret, R., Agosto-Rivera, J.L., Giray, T., 2022. Honeybee re-identification in video: New datasets and impact of self-supervision, in: *VISIGRAPP (5: VISAPP)*, pp. 517–525.
- Chen, Y.S., Kuan, C.Y., Hsu, J.T., Lin, T.T., 2021. Lightweight cow face recognition algorithm based on few-shot learning for edge computing application, in: *2021 ASABE Annual International Virtual Meeting*, American Society of Agricultural and Biological Engineers. p. 1.
- Costa, S., Pedro, S., Lourenço, H., Batista, I., Teixeira, B., Bandarra, N.M., Murta, D., Nunes, R., Pires, C., 2020. Evaluation of *tenebrio molitor* larvae as an alternative food source. *NFS journal* 21, 57–64.
- Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V., 2016. Domain-adversarial training of neural networks. *The journal of machine learning research* 17, 2096–2030.
- Gong, S., Xiang, T., Gong, S., Xiang, T., 2011. *Person re-identification*. Springer.
- Grau, T., Vilcinskis, A., Joop, G., 2017. Sustainable farming of the mealworm *tenebrio molitor* for the production of food and feed. *Zeitschrift für Naturforschung C* 72, 337–349.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.
- Heckmann, L.H., Andersen, J.L., Gianotten, N., Calis, M., Fischer, C.H., Calis, H., 2018. Sustainable mealworm production for feed and food. *Edible insects in sustainable food systems*, 321–328.
- Hermans, A., Beyer, L., Leibe, B., 2017. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*.
- Jocher, G., Chaurasia, A., Qiu, J., 2023. Ultralytics yolov8. URL: <https://github.com/ultralytics/ultralytics>.
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM* 60, 84–90.
- Lahiri, M., Tantipathanandh, C., Warungu, R., Rubenstein, D.I., Berger-Wolf, T.Y., 2011. Biometric animal databases from field photographs: identification of individual zebra in the wild, in: *Proceedings of the 1st ACM international conference on multimedia retrieval*, pp. 1–8.
- Li, S., Li, J., Tang, H., Qian, R., Lin, W., 2019. Atrw: a benchmark for amur tiger re-identification in the wild. *arXiv preprint arXiv:1906.05586*.
- Van der Maaten, L., Hinton, G., 2008. Visualizing data using t-sne. *Journal of machine learning research* 9.
- Majewski, P., Mrzygłód, M., Lampa, P., Burduk, R., Reiner, J., 2024. Monitoring the growth of insect larvae using a regression convolutional neural network and knowledge transfer. *Engineering Applications of Artificial Intelligence* 127, 107358.
- Majewski, P., Zapotoczny, P., Lampa, P., Burduk, R., Reiner, J., 2022. Multipurpose monitoring system for edible insect breeding based on machine learning. *Scientific Reports* 12, 7892.
- Mancini, S., Sogari, G., Espinosa Diaz, S., Menozzi, D., Paci, G., Moruzzo, R., 2022. Exploring the future of edible insects in europe. *Foods* 11, 455.
- Morales-Ramos, J.A., Kelstrup, H.C., Rojas, M.G., Emery, V., 2019. Body mass increase induced by eight years of artificial selection in the yellow mealworm (coleoptera: Tenebrionidae) and life history trade-offs. *Journal of Insect Science* 19, 4.
- Movshovitz-Attias, Y., Toshev, A., Leung, T.K., Ioffe, S., Singh, S., 2017. No fuss distance metric learning using proxies, in: *Proceedings of the IEEE international conference on computer vision*, pp. 360–368.

## Phenotyping of *Tenebrio Molitor* beetles

- Murali, N., Schneider, J., Levine, J., Taylor, G., 2019. Classification and re-identification of fruit fly individuals across days with convolutional neural networks, in: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE. pp. 570–578.
- Musgrave, K., Belongie, S., Lim, S.N., 2020a. A metric learning reality check, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16, Springer. pp. 681–699.
- Musgrave, K., Belongie, S.J., Lim, S.N., 2020b. Pytorch metric learning. ArXiv abs/2008.09164.
- Neethirajan, S., Kemp, B., 2021. Digital livestock farming. *Sensing and Bio-Sensing Research* 32, 100408.
- Padilla, R., Netto, S.L., Da Silva, E.A., 2020. A survey on performance metrics for object-detection algorithms, in: 2020 international conference on systems, signals and image processing (IWSSIP), IEEE. pp. 237–242.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520.
- Schneider, S., Taylor, G.W., Linqvist, S., Kremer, S.C., 2019. Past, present and future approaches using computer vision for animal re-identification from camera trap data. *Methods in Ecology and Evolution* 10, 461–470.
- Song, J.H., Chang, G.D., Ji, S., Kim, S.Y., Kim, W., 2022. Selective breeding and characterization of a black mealworm strain of *tenebrio molitor* linnaeus (coleoptera: Tenebrionidae). *Journal of Asia-Pacific Entomology* 25, 101978.
- Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., Wei, Y., 2020. Circle loss: A unified perspective of pair similarity optimization, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 6398–6407.
- Tan, M., Le, Q., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR. pp. 6105–6114.
- Tang, C., Yang, D., Liao, H., Sun, H., Liu, C., Wei, L., Li, F., 2019. Edible insects as a food source: a review. *Food Production, Processing and Nutrition* 1, 1–13.
- Tausch, F., Stock, S., Fricke, J., Klein, O., 2020. Bumblebee re-identification dataset, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, pp. 35–37.
- Van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T., 2014. scikit-image: image processing in python. *PeerJ* 2, e453.
- Wang, M., Larsen, M.L., Liu, D., Winters, J.F., Rault, J.L., Norton, T., 2022. Towards re-identification for long-term tracking of group housed pigs. *Biosystems Engineering* 222, 71–81.
- Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R., 2019. Multi-similarity loss with general pair weighting for deep metric learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 5022–5030.
- Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 600–612.



# Bibliography

---

- [1] Aharon, S., Louis-Dupont, Ofri Masad, Yurkova, K., Lotem Fridman, Lkdci, Khvedchenya, E., Rubin, R., Bagrov, N., Tymchenko, B., Keren, T., Zhilko, A., and Eran-Deci (2021). Super-gradients. → [p5]
- [2] Ahmed, S., Alshater, M. M., El Ammari, A., and Hammami, H. (2022). Artificial intelligence and machine learning in finance: A bibliometric review. *Research in International Business and Finance*, 61:101646. → [p1]
- [3] Alves, T. S., Pinto, M. A., Ventura, P., Neves, C. J., Biron, D. G., Junior, A. C., De Paula Filho, P. L., and Rodrigues, P. J. (2020). Automatic detection and classification of honey bee comb cells using deep learning. *Computers and Electronics in Agriculture*, 170:105244. → [p4]
- [4] Anagnostopoulou, D., Retsinas, G., Efthymiou, N., Filntisis, P., and Maragos, P. (2023). A realistic synthetic mushroom scenes dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6282–6289. → [p8]
- [5] Baur, A., Koch, D., Gatternig, B., and Delgado, A. (2022). Noninvasive monitoring system for tenebrio molitor larvae based on image processing with a watershed algorithm and a neural net approach. *Journal of Insects as Food and Feed*, 8(8):913–920. → [p5]
- [6] Beck, M. A., Liu, C.-Y., Bidinosti, C. P., Henry, C. J., Godee, C. M., and Ajmani, M. (2020). An embedded system for the automated generation of labeled plant images to enable machine learning applications in agriculture. *Plos one*, 15(12):e0243923. → [p11]
- [7] Bergmann, P., Fauser, M., Sattlegger, D., and Steger, C. (2020). Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4183–4192. → [p11]
- [8] Bissoto, A., Valle, E., and Avila, S. (2021). Gan-based data augmentation and anonymization for skin-lesion analysis: A critical review. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1847–1856. → [p8]
- [9] Bjerge, K., Frigaard, C. E., Mikkelsen, P. H., Nielsen, T. H., Misbih, M., and Kryger, P. (2019). A computer vision system to monitor the infestation level of varroa destructor in a honeybee colony. *Computers and Electronics in Agriculture*, 164:104898. → [p2], [p4], [p7]
- [10] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*. → [p4]



- [11] Borlinghaus, P., Tausch, F., and Rettenberger, L. (2023). A purely visual re-id approach for bumblebees (*bombus terrestris*). *Smart Agricultural Technology*, 3:100135. → [p4], [p14]
- [12] Bota, P. J., Wang, C., Fred, A. L., and Da Silva, H. P. (2019). A review, current challenges, and future possibilities on emotion recognition using machine learning and physiological signals. *IEEE access*, 7:140990–141020. → [p1]
- [13] Bozek, K., Hebert, L., Mikheyev, A. S., and Stephens, G. J. (2018). Towards dense object tracking in a 2d honeybee hive. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4185–4193. → [p3]
- [14] Bozek, K., Hebert, L., Portugal, Y., Mikheyev, A. S., and Stephens, G. J. (2021). Markerless tracking of an entire honey bee colony. *Nature communications*, 12(1):1733. → [p3]
- [15] Cang, Y., He, H., and Qiao, Y. (2019). An intelligent pig weights estimate method based on deep learning in sow stall environments. *IEEE Access*, 7:164867–164875. → [p3], [p11], [p12]
- [16] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299. → [p4]
- [17] Carraro, A., Sozzi, M., and Marinello, F. (2023). The segment anything model (sam) for accelerating the smart farming revolution. *Smart Agricultural Technology*, 6:100367. → [p11]
- [18] Cejrowski, T. and Szymański, J. (2021). Buzz-based honeybee colony fingerprint. *Computers and Electronics in Agriculture*, 191:106489. → [p2]
- [19] Cejrowski, T., Szymański, J., and Logofătu, D. (2020). Buzz-based recognition of the honeybee colony circadian rhythm. *Computers and Electronics in Agriculture*, 175:105586. → [p2]
- [20] Chan, J., Carrión, H., Mégret, R., Agosto-Rivera, J. L., and Giray, T. (2022). Honeybee re-identification in video: New datasets and impact of self-supervision. In *VISIGRAPP (5: VISAPP)*, pages 517–525. → [p5], [p14]
- [21] Chen, W., Liu, M., Zhao, C., Li, X., and Wang, Y. (2024a). Mtd-yolo: Multi-task deep convolutional neural network for cherry tomato fruit bunch maturity detection. *Computers and Electronics in Agriculture*, 216:108533. → [p2]
- [22] Chen, Z., Yang, R., Zhang, S., Norton, T., Shen, M., Wang, F., and Lu, M. (2024b). Recognizing pawing behavior of prepartum doe using semantic segmentation and motion history image (mhi) features. *Expert Systems with Applications*, 242:122829. → [p3]
- [23] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20:273–297. → [p3], [p10]

- 
- [24] Dadboud, F., Patel, V., Mehta, V., Bolic, M., and Mantegh, I. (2021). Single-stage uav detection and classification with yolov5: Mosaic data augmentation and panet. In *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8. IEEE. → [p7]
- [25] Dembski, J. and Szymański, J. (2020). Weighted clustering for bees detection on video images. In *Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part V 20*, pages 453–466. Springer. → [p4]
- [26] dos Santos Ferreira, A., Freitas, D. M., da Silva, G. G., Pistori, H., and Folhes, M. T. (2019). Unsupervised deep learning and semi-automatic data labeling in weed discrimination. *Computers and Electronics in Agriculture*, 165:104963. → [p11]
- [27] Duda, R. O. and Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15. → [p4]
- [28] Gao, J., Yang, Y., Lin, P., and Park, D. S. (2018). Computer vision in healthcare applications. *Journal of healthcare engineering*, 2018. → [p1]
- [29] García, R., Aguilar, J., Toro, M., Pinto, A., and Rodríguez, P. (2020). A systematic literature review on the use of machine learning in precision livestock farming. *Computers and Electronics in Agriculture*, 179:105826. → [p1]
- [30] Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.-Y., Cubuk, E. D., Le, Q. V., and Zoph, B. (2021). Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2918–2928. → [p21]
- [31] Gomes, P. A., Suhara, Y., Nunes-Silva, P., Costa, L., Arruda, H., Venturieri, G., Imperatriz-Fonseca, V. L., Pentland, A., Souza, P. d., and Pessin, G. (2020). An amazon stingless bee foraging activity predicted using recurrent artificial neural networks and attribute selection. *Scientific reports*, 10(1):9. → [p12]
- [32] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27. → [p1]
- [33] Graczyk, K. M., Pawłowski, J., Majchrowska, S., and Golan, T. (2022). Self-normalized density map (sndm) for counting microbiological objects. *Scientific Reports*, 12(1):10583. → [p9]
- [34] Gu, Q., Huang, F., Lou, W., Zhu, Y., Hu, H., Zhao, Y., Zhou, H., and Zhang, X. (2024). Unmanned aerial vehicle-based assessment of rice leaf chlorophyll content dynamics across genotypes. *Computers and Electronics in Agriculture*, 221:108939. → [p2]
- [35] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. → [p4]
-

- [36] Hong, W., Xu, B., Chi, X., Cui, X., Yan, Y., and Li, T. (2020). Long-term and extensive monitoring for bee colonies based on internet of things. *IEEE Internet of Things Journal*, 7(8):7148–7155. → [p2]
- [37] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*. → [p4]
- [38] Inoue, N., Furuta, R., Yamasaki, T., and Aizawa, K. (2018). Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009. → [p8]
- [39] Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3304–3311. IEEE. → [p3]
- [40] Jocher, G., Chaurasia, A., and Qiu, J. (2023). Ultralytics yolov8. → [p5], [p8], [p11]
- [41] Keys, P. W., Barnes, E. A., and Carter, N. H. (2021). A machine-learning approach to human footprint index estimation with applications to sustainable development. *Environmental Research Letters*, 16(4):044061. → [p11]
- [42] Khodabandeh, M., Vahdat, A., Ranjbar, M., and Macready, W. G. (2019). A robust learning approach to domain adaptive object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 480–490. → [p9], [p10]
- [43] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25. → [p1], [p10]
- [44] Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97. → [p4]
- [45] Larsen, M. L., Wang, M., and Norton, T. (2021). Information technologies for welfare monitoring in pigs and their relation to welfare quality®. *Sustainability*, 13(2):692. → [p3]
- [46] Larsen, M. L., Wang, M., Willems, S., Liu, D., and Norton, T. (2023). Automatic detection of locomotor play in young pigs: A proof of concept. *Biosystems Engineering*, 229:154–166. → [p3]
- [47] Lea, C., Vidal, R., Reiter, A., and Hager, G. D. (2016). Temporal convolutional networks: A unified approach to action segmentation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8–10 and 15–16, 2016, Proceedings, Part III 14*, pages 47–54. Springer. → [p4]
- [48] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444. → [p1]

- 
- [49] Li, K. and Malik, J. (2016). Amodal instance segmentation. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 677–693. Springer. → [p10]
- [50] Li, Q., Ma, W., Li, H., Zhang, X., Zhang, R., and Zhou, W. (2024). Cotton-yolo: Improved yolov7 for rapid detection of foreign fibers in seed cotton. *Computers and Electronics in Agriculture*, 219:108752. → [p10]
- [51] Liao, J., Chen, M., Zhang, K., Zhou, H., Zou, Y., Xiong, W., Zhang, S., Kuang, F., and Zhu, D. (2024). Sc-net: A new strip convolutional network model for rice seedling and weed segmentation in paddy field. *Computers and Electronics in Agriculture*, 220:108862. → [p2]
- [52] Liu, C., Feng, Q., Sun, Y., Li, Y., Ru, M., and Xu, L. (2023). Yolactfusion: An instance segmentation method for rgb-nir multimodal image fusion based on an attention mechanism. *Computers and Electronics in Agriculture*, 213:108186. → [p2]
- [53] Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth IEEE international conference on data mining*, pages 413–422. IEEE. → [p4]
- [54] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee. → [p3]
- [55] Lu, C.-Y., Rustia, D. J. A., and Lin, T.-T. (2019). Generative adversarial network based image augmentation for insect pest classification enhancement. *IFAC-PapersOnLine*, 52(30):1–5. → [p7]
- [56] Mao, A., Huang, E., Wang, X., and Liu, K. (2023). Deep learning-based animal activity recognition with wearable sensors: Overview, challenges, and future directions. *Computers and Electronics in Agriculture*, 211:108043. → [p2], [p3]
- [57] Marstaller, J., Tausch, F., and Stock, S. (2019). Deepbees-building and scaling convolutional neuronal nets for fast and large-scale visual monitoring of bee hives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0. → [p4], [p9], [p11]
- [58] McInnes, L., Healy, J., Astels, S., et al. (2017). hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205. → [p5]
- [59] Meyer, L., Gilson, A., Scholz, O., and Stamminger, M. (2023). Cherrypicker: Semantic skeletonization and topological reconstruction of cherry trees. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 6244–6253. → [p3]
- [60] Mirbod, O. and Choi, D. (2023). Synthetic data-driven ai using mixture of rendered and real imaging data for strawberry yield estimation. In *2023 ASABE Annual International Meeting*, page 1. American Society of Agricultural and Biological Engineers. → [p8]
-

- [61] Morales-Ramos, J. A., Kelstrup, H. C., Rojas, M. G., and Emery, V. (2019). Body mass increase induced by eight years of artificial selection in the yellow mealworm (coleoptera: Tenebrionidae) and life history trade-offs. *Journal of Insect Science*, 19(2):4. → [p12]
- [62] Murali, N., Schneider, J., Levine, J., and Taylor, G. (2019). Classification and re-identification of fruit fly individuals across days with convolutional neural networks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 570–578. IEEE. → [p14]
- [63] Nasiri, A., Yoder, J., Zhao, Y., Hawkins, S., Prado, M., and Gan, H. (2022). Pose estimation-based lameness recognition in broiler using cnn-lstm network. *Computers and Electronics in Agriculture*, 197:106931. → [p3]
- [64] Ngai, E. W. and Wu, Y. (2022). Machine learning in marketing: A literature review, conceptual framework, and research agenda. *Journal of Business Research*, 145:35–48. → [p1]
- [65] Ngo, T. N., Rustia, D. J. A., Yang, E.-C., and Lin, T.-T. (2021a). Automated monitoring and analyses of honey bee pollen foraging behavior using a deep learning-based imaging system. *Computers and Electronics in Agriculture*, 187:106239. → [p4], [p12]
- [66] Ngo, T.-N., Rustia, D. J. A., Yang, E.-C., and Lin, T.-T. (2021b). Honey bee colony population daily loss rate forecasting and an early warning method using temporal convolutional networks. *Sensors*, 21(11):3900. → [p4]
- [67] Ngo, T. N., Wu, K.-C., Yang, E.-C., and Lin, T.-T. (2019). A real-time imaging system for multiple honey bee tracking and activity monitoring. *Computers and Electronics in Agriculture*, 163:104841. → [p4]
- [68] Panda, S. K., Lee, Y., and Jawed, M. K. (2023). Agronav: Autonomous navigation framework for agricultural robots and vehicles using semantic segmentation and semantic line detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 6272–6281. → [p2]
- [69] Papadopoulos, A.-M., Melissas, P., Kastellos, A., Katranitsiotis, P., Zapparas, P., Stavridis, K., and Daras, P. (2024). Tenebriovision: A fully annotated dataset of tenebrio molitor larvae worms in a controlled environment for accurate small object detection and segmentation. In *Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM*, pages 187–196. INSTICC, SciTePress. → [p5]
- [70] Pawłowski, J., Majchrowska, S., and Golan, T. (2022). Generation of microbial colonies dataset with deep learning style transfer. *Scientific Reports*, 12(1):5212. → [p7]
- [71] Rai, R., Tiwari, M. K., Ivanov, D., and Dolgui, A. (2021). Machine learning in manufacturing and industry 4.0 applications. → [p1]



- 
- [72] Ratnayake, M. N., Amarathunga, D. C., Zaman, A., Dyer, A. G., and Dorin, A. (2023). Spatial monitoring and insect behavioural analysis using computer vision for precision pollination. *International Journal of Computer Vision*, 131(3):591–606. → [p4]
- [73] Ratnayake, M. N., Dyer, A. G., and Dorin, A. (2021a). Towards computer vision and deep learning facilitated pollination monitoring for agriculture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2921–2930. → [p4]
- [74] Ratnayake, M. N., Dyer, A. G., and Dorin, A. (2021b). Tracking individual honeybees among wildflower clusters with computer vision-facilitated pollinator monitoring. *Plos one*, 16(2):e0239504. → [p4]
- [75] Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*. → [p4]
- [76] Rodríguez, I., Branson, K., Acuña, E., Agosto-Rivera, J., Giray, T., and Mégret, R. (2018). Honeybee detection and pose estimation using convolutional neural networks. In *Congres Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP)*. → [p4]
- [77] Rodriguez, I. F., Chan, J., Alvarez Rios, M., Branson, K., Agosto-Rivera, J. L., Giray, T., and Mégret, R. (2022). Automated video monitoring of unmarked and marked honey bees at the hive entrance. *Frontiers in Computer Science*, 3:769338. → [p4], [p11]
- [78] Rodriguez, I. F., Megret, R., Acuna, E., Agosto-Rivera, J. L., and Giray, T. (2018a). Recognition of pollen-bearing bees from video using convolutional neural network. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 314–322. IEEE. → [p4]
- [79] Rodriguez, I. F., Mégret, R., Egnor, R., Branson, K., Agosto, J. L., Giray, T., and Acuna, E. (2018b). Multiple insect and animal tracking in video using part affinity fields. In *Workshop Visual observation and analysis of Vertebrate And Insect Behavior (VAIB) at International Conference on Pattern Recognition (ICPR)*. → [p4]
- [80] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer. → [p3]
- [81] Sadeghi-Tehran, P., Sabermanesh, K., Virlet, N., and Hawkesford, M. J. (2017). Automated method to determine two critical growth stages of wheat: heading and flowering. *Frontiers in Plant Science*, 8:233406. → [p3]
- [82] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. → [p4]
- [83] Sledević, T. and Plonis, D. (2023). Toward bee behavioral pattern recognition on hive entrance using yolov8. In *2023 IEEE 10th Jubilee Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, pages 1–4. IEEE. → [p5]

- [84] Smorodov, A. (2017). Multiple object tracker. → [p4]
- [85] Stojnić, V., Risojević, V., and Pilipović, R. (2018). Detection of pollen bearing honey bees in hive entrance images. In *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–4. IEEE. → [p3]
- [86] Szturo, K. and Szczypiński, P. M. (2017). Ontology based expert system for barley grain classification. In *2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pages 360–364. IEEE. → [p10]
- [87] Tashakkori, R., Hamza, A. S., and Crawford, M. B. (2021). Beemon: An iot-based beehive monitoring system. *Computers and Electronics in Agriculture*, 190:106427. → [p2]
- [88] Tausch, F., Stock, S., Fricke, J., and Klein, O. (2020). Bumblebee re-identification dataset. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 35–37. → [p4]
- [89] Tausch, F., Wagner, J., and Klaus, S. (2023). Pollinators as data collectors: Estimating floral diversity with bees and computer vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 643–650. → [p5]
- [90] Tian, H., Wang, T., Liu, Y., Qiao, X., and Li, Y. (2020). Computer vision technology in agricultural automation—a review. *Information Processing in Agriculture*, 7(1):1–19. → [p1]
- [91] Tian, Y., Wang, S., Li, E., Yang, G., Liang, Z., and Tan, M. (2023). Md-yolo: Multi-scale dense yolo for small target pest detection. *Computers and Electronics in Agriculture*, 213:108233. → [p11]
- [92] Toda, Y., Okura, F., Ito, J., Okada, S., Kinoshita, T., Tsuji, H., and Saisho, D. (2020). Training instance segmentation neural network with synthetic datasets for crop seed phenotyping. *Communications biology*, 3(1):173. → [p8], [p21]
- [93] Tsai, Y.-C., Hsu, J.-T., Ding, S.-T., Rustia, D. J. A., and Lin, T.-T. (2020). Assessment of dairy cow heat stress by monitoring drinking behaviour using an embedded imaging system. *biosystems engineering*, 199:97–108. → [p3]
- [94] Tuan, S.-A., Rustia, D. J. A., Hsu, J.-T., and Lin, T.-T. (2022). Frequency modulated continuous wave radar-based system for monitoring dairy cow respiration rate. *Computers and Electronics in Agriculture*, 196:106913. → [p3]
- [95] Unold, O., Nikodem, M., Piasecki, M., Szyk, K., Maciejewski, H., Bawiec, M., Dobrowolski, P., and Zdunek, M. (2020). Iot-based cow health monitoring system. In *International Conference on Computational Science*, pages 344–356. Springer. → [p2]
- [96] Valloli, V. K. and Mehta, K. (2019). W-net: Reinforced u-net for density map estimation. *arXiv preprint arXiv:1903.11249*. → [p9]

- 
- [97] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30. → [p1]
- [98] Wan, L., Liu, Y., He, Y., and Cen, H. (2023). Prior knowledge and active learning enable hybrid method for estimating leaf chlorophyll content from multi-scale canopy reflectance. *Computers and Electronics in Agriculture*, 214:108308. → [p10]
- [99] Wang, H., Pan, X., Zhu, Y., Li, S., and Zhu, R. (2024). Maize leaf disease recognition based on tc-mrsn model in sustainable agriculture. *Computers and Electronics in Agriculture*, 221:108915. → [p2]
- [100] Wang, L. and Yoon, K.-J. (2021). Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE transactions on pattern analysis and machine intelligence*, 44(6):3048–3068. → [p11]
- [101] Wang, M., Larsen, M., Bayer, F., Maschat, K., Baumgartner, J., Rault, J.-L., Norton, T., et al. (2021). A pca-based frame selection method for applying cnn and lstm to classify postural behaviour in sows. *Computers and Electronics in Agriculture*, 189:106351. → [p3]
- [102] Wang, M., Li, X., Larsen, M. L., Liu, D., Rault, J.-L., and Norton, T. (2023a). A computer vision-based approach for respiration rate monitoring of group housed pigs. *Computers and Electronics in Agriculture*, 210:107899. → [p3]
- [103] Wang, X., Wang, Y., Zhao, J., and Niu, J. (2023b). Eca-convnext: A rice leaf disease identification model based on convnext. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 6235–6243. → [p2]
- [104] Wang, Y., Deng, X., Luo, J., Li, B., and Xiao, S. (2023c). Cross-task feature enhancement strategy in multi-task learning for harvesting sichuan pepper. *Computers and Electronics in Agriculture*, 207:107726. → [p11]
- [105] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612. → [p9]
- [106] Westwańska, W. W. and Respondek, J. S. (2019). Counting instances of objects in color images using u-net network on example of honey bees. In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 87–90. IEEE. → [p4], [p9]
- [107] Xiang, S., Wang, S., Xu, M., Wang, W., and Liu, W. (2023). Yolo pod: a fast and accurate multi-task model for dense soybean pod counting. *Plant methods*, 19(1):8. → [p11], [p12]
- [108] Xu, A. and Buchanan, R. L. (2020). Evaluation of a hybrid in-field sampling method for the detection of pathogenic bacteria through consideration of a priori knowledge of factors related to non-random contamination. *Food microbiology*, 89:103412. → [p10]
-

- [109] Yang, D., Yang, H., Liu, D., and Wang, X. (2024). Research on automatic 3d reconstruction of plant phenotype based on multi-view images. *Computers and Electronics in Agriculture*, 220:108866. → [p3]
- [110] Yang, F., Zhang, D., Zhang, Y., Zhang, Y., Han, Y., Zhang, Q., Zhang, Q., Zhang, C., Liu, Z., and Wang, K. (2023). Prediction of corn variety yield with attribute-missing data via graph neural network. *Computers and Electronics in Agriculture*, 211:108046. → [p3]
- [111] Ye, Y., Chen, Y., and Xiong, S. (2024). Field detection of pests based on adaptive feature fusion and evolutionary neural architecture search. *Computers and Electronics in Agriculture*, 221:108936. → [p2]
- [112] Zantalis, F., Koulouras, G., Karabetsos, S., and Kandris, D. (2019). A review of machine learning and iot in smart transportation. *Future Internet*, 11(4):94. → [p1]
- [113] Zhang, J., Zhuang, Y., Ji, H., and Teng, G. (2021). Pig weight and body size estimation using a multiple output regression convolutional neural network: A fast and fully automatic method. *Sensors*, 21(9):3218. → [p2], [p3], [p11]
- [114] Zhang, K., Han, S., Wu, J., Cheng, G., Wang, Y., Wu, S., and Liu, J. (2023). Early lameness detection in dairy cattle based on wearable gait analysis using semi-supervised lstm-autoencoder. *Computers and Electronics in Agriculture*, 213:108252. → [p3]
- [115] Zhang, X., Lu, X., Zhang, Z., Yang, G., He, Y., and Fang, H. (2024). Simultaneous detection of reference lines in paddy fields using a machine vision-based framework. *Computers and Electronics in Agriculture*, 221:108923. → [p2]
- [116] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., and Wang, X. (2022). Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*, pages 1–21. Springer. → [p5]
- [117] Zhao, R., Zhu, Y., and Li, Y. (2023). Cla: A self-supervised contrastive learning method for leaf disease identification with domain adaptation. *Computers and Electronics in Agriculture*, 211:107967. → [p10]
- [118] Zhao, T., Shen, Z., Zou, H., Zhong, P., and Chen, Y. (2022). Unsupervised adversarial domain adaptation based on interpolation image for fish detection in aquaculture. *Computers and Electronics in Agriculture*, 198:107004. → [p10]
- [119] Zheng, Z., Zhang, X., Qin, L., Yue, S., and Zeng, P. (2023). Cows' legs tracking and lameness detection in dairy cattle using video analysis and siamese neural networks. *Computers and Electronics in Agriculture*, 205:107618. → [p3]
- [120] Zhu, S., Cui, N., Guo, L., Jin, H., Jin, X., Jiang, S., Wu, Z., Lv, M., Chen, F., Liu, Q., et al. (2024). Enhancing precision of root-zone soil moisture content prediction in a kiwifruit orchard using uav multi-spectral image features and ensemble learning. *Computers and Electronics in Agriculture*, 221:108943. → [p2]

# List of Abbreviations

---

**CNN** convolutional neural network

**CV** computer vision

**HB** honeybee

**HDBSCAN** hierarchical density-based spatial clustering of applications with noise

**K-Means** clustering algorithm

**Mask R-CNN** mask region-based convolutional neural network, instance segmentation model

**ML** machine learning

**MLP** multilayer perceptron

**MW** mealworm (*Tenebrio Molitor*)

**NIR** near infrared

**PA** precision agriculture

**PB** precision beekeeping

**PCA** principal component analysis

**PIF** precision insect farming

**PLF** precision livestock farming

**RegCNN** regression convolutional neural network

**ROI** region of interest

**SIFT** scale-invariant feature transform

**SVM** support vector machines

**TSNE** t-distributed stochastic neighbor embedding (dimensionality reduction technique)



*List of Abbreviations*

---

**U-Net** semantic segmentation model

**VLAD** vector of locally aggregated descriptors

**YOLO** You Only Look Once, group of object detection models