



Politechnika Wrocławska

FIELD OF SCIENCE: Engineering and Technology

DISCIPLINE OF SCIENCE: Information and Communication Technology

DOCTORAL DISSERTATION

Deep learning methods for computer vision-based industrial inspection applications

mgr inż. Przemysław Dolata

Supervisor:

dr hab. Maciej Zięba, prof. uczelni

Keywords: deep learning, computer vision, image classification, semantic segmentation, object detection, instance segmentation

WROCŁAW 2025

Table of contents

Abstract	5
Streszczenie	7
List of publications	9
1 Introduction	11
1.1 Background	11
1.2 Research scope	12
2 Multi-view classification	15
2.1 Context and research scope	15
2.2 Multi-view information fusion	16
2.3 Viewpoint-specific feature extraction	18
3 Fast semantic segmentation	21
3.1 Context and research scope	21
3.2 U-Net architecture optimization	22
4 Instance segmentation for object dimensioning	25
4.1 Context and research scope	25
4.2 3D size estimation from a single image	27
4.3 Training data for amodal instance segmentation	29
4.4 Object detection in high-density images	31
5 Conclusions	35
Acknowledgements	37
Bibliography	39
Full texts of publications	43
1 Double-stream Convolutional Neural Networks for Machine Vision Inspection of Natural Products	45
2 Barley Variety Recognition with Viewpoint-aware Double-stream Convolutional Neural Networks	63
3 Improving Real-Time Performance of U-Nets for Machine Vision in Laser Process Control	69
4 Instance Segmentation of Root Crops and Simulation-based Learning to Estimate their Physical Dimensions for On-line Machine Vision Yield Monitoring	75
5 Amodal Instance Segmentation for Mealworm Growth Monitoring Using Synthetic Training Images	87
6 Direct Detection of Elongated Objects Geometry via a Centerline-based Representation	107

Abstract

Computer vision is finding increasingly more uses in the automation of industrial inspection tasks. Methods based on deep learning have unlocked even more application avenues, previously inaccessible to classical approaches. The concept of transfer learning and the growing availability of large, high-performance vision models have made it easier than ever before to deploy computer vision systems in practice. But despite that progress, there still exist domains where off-the-shelf deep learning models do not adapt with similarly superlative performance, necessitating further research.

The goal of this thesis is to develop specialized deep learning methods for computer vision-based industrial inspection applications in domains where the standard transfer learning approach does not yield satisfying results. Such application domains include agriculture, food processing, or certain process control scenarios. Due to the unique requirements or limitations in those domains, the generic deep models either do not transfer well or do not fully solve the vision problems. In order to achieve the goal of the thesis, three research questions were formulated and answered in a series of six publications.

The first research question concerned the design of convolutional neural network-based image classifiers for the special *multi-view* classification setting, in which the sample to be classified comprises a tuple of images. Such scenarios arise in certain quality inspection problems, where objects need to be imaged using multiple cameras, for example, in the food industry. Standard classification models are not able to reason from multiple images without substantial, non-trivial modifications. This question was addressed in Publications 1 and 2, discussing the design and training of double-stream convolutional neural networks. Publication 1 explores methods of fusing information from multiple inputs, while Publication 2 proposes a special case of a viewpoint-specific classification model.

The second research question was related to the optimization of the U-Net for semantic segmentation in low-latency applications. Certain tasks of industrial process control, for example, the monitoring of laser processes in additive manufacturing, could be automated using computer vision. But those tasks require models with exceptionally high inference speeds, since real-time processing is crucial for control. The U-Net, while offering high segmentation quality, is too computationally inefficient for this purpose. An architectural optimization method was proposed in Publication 3, which was awarded the Best Paper award at FedCSIS 2019.

The final research question involved the applications of deep learning-based instance segmentation methods for the purpose of size estimation of small objects in dense scenes from single images. The small size of objects, their large number and high density, often causing occlusions, and the need to accurately estimate their size, even from non-3D images, are the frequent challenges in industrial quality inspection, for example, in agriculture or food processing. At the same time, generic deep models trained on large datasets struggle significantly in those conditions. Three sub-questions were formulated in order to study those issues, and three Publications (4, 5, and 6) were presented to address them. The first sub-question, addressed in Publication 4, concerned the problem of size estimation of small three-dimensional objects in single images without depth information. The second sub-question asked for a method of training *amodal* instance segmentation models, that is, models capable of segmenting occluded objects, without amodal data annotations; such an approach was presented in Publication 5. The third sub-question was related to improving the detection of small objects in high-density images, with a novel detection method proposed in Publication 6.

In total, this thesis summarizes findings from six Publications, all of which I am the first author of. To date, those Publications have been cited 37 times, confirming their academic impact. Additionally, four of them were a result of research activity funded by research and development grants, highlighting their practical value in the field of industrial inspection.

Streszczenie

Widzenie komputerowe znajduje liczne zastosowania w zadaniach automatycznej inspekcji w różnych gałęziach przemysłu. Metody oparte o uczenie głębokie umożliwiły stosowanie inspekcji wizyjnej w scenariuszach, w których techniki klasyczne okazywały się niewystarczające. Idea uczenia przez transfer oraz rosnąca dostępność głębokich modeli wizyjnych o wysokiej jakości uczyniły wdrażanie złożonych systemów wizyjnych jeszcze łatwiejszym. Jednak pomimo postępu, standardowe modele nadal nie osiągają podobnie spektakularnych wyników w pewnych dziedzinach. Oznacza to, że aby z powodzeniem wdrażać tam systemy wizyjne oparte o modele głębokie, konieczne są dalsze badania.

Celem niniejszej rozprawy jest opracowanie metod uczenia głębokiego dla wybranych zastosowań inspekcji wizyjnej, w szczególności takich, gdzie standardowe podejście oparte o uczenie przez transfer nie okazało się skuteczne lub możliwe. Przykłady takich zagadnień pojawiają się w dziedzinach takich jak rolnictwo precyzyjne, przetwórstwie żywności, a także pewnych scenariuszach sterowania procesami przemysłowymi. Wyjątkowe uwarunkowania i wymagania stawiane systemom wizyjnym w tych dziedzinach sprawiają, że ogólne modele dostępne w literaturze albo nie transferują się do nich z powodzeniem, albo nie rozwiązują w całości problemu wizyjnego. Aby osiągnąć cel rozprawy, w jej toku postawione zostały trzy pytania badawcze, odpowiedzi na które znajdują się w cyklu sześciu publikacji naukowych.

Pierwsze pytanie badawcze dotyczyło opracowania klasyfikatorów opartych o konwolucyjne sieci neuronowe do szczególnego przypadku klasyfikacji *wielowidkowej*, w której próbkę podlegającą klasyfikacji opisuje kilka obrazów. Taką formę zadania przyjmuje często automatyczna inspekcja jakości, podczas której obiekty muszą być obrazowane z wielu stron. Przykłady tak sformułowanej inspekcji często spotykane są w przetwórstwie żywności. Struktura ogólnych modeli głębokich do klasyfikacji nie pozwala na użycie ich do wnioskowania z wielu obrazów bez znaczących modyfikacji. Odpowiedź na to pytanie znajduje się w Publikacjach 1 i 2, które omawiają budowę i uczenie dwustrumieniowych konwolucyjnych sieci neuronowych. W Publikacji 1 zbadane zostały metody łączenia informacji z wielu obrazów wejściowych, a w Publikacji 2 zaproponowany został szczególny przypadek modelu klasyfikacyjnego o specjalizowanych strumieniach ekstrakcji cech.

Drugie pytanie badawcze związane było z optymalizacją struktury sieci U-Net do zadań segmentacji semantycznej wymagających przetwarzania o niskim opóźnieniu. Niektóre zadania sterowania procesami przemysłowymi, na przykład monitorowanie procesów laserowych w obróbce addytywnej, mogłyby być automatyzowane za pomocą widzenia komputerowego. Jednak w zadaniach tych niezbędne jest przetwarzanie w czasie rzeczywistym, dlatego wymagają one modeli o bardzo dużej szybkości przetwarzania. U-Net, choć oferuje wysoką jakość segmentacji, cechuje wysoka złożoność obliczeniowa, sprawiając, że model ten jest niewystarczająco wydajny do tego celu. W Publikacji 3 zaproponowana została metoda optymalizacji architektury tego modelu, pozwalająca na znaczące skrócenie czasu inferencji przy jednoczesnym zachowaniu jakości predykcji. Publikacja ta została nagrodzona nagrodą *Best Paper* na konferencji FedCSIS 2019.

Ostatnie pytanie badawcze dotyczyło zastosowań metod uczenia głębokiego do segmentacji instancyjnej dla celów estymacji wymiarów gęsto rozmieszczonych małych obiektów na podstawie pojedynczych obrazów. Niewielki rozmiar obiektów, ich duża liczba i bliskość, często powodująca powstawanie okluzji, w połączeniu z potrzebą dokładnego szacowania ich wymiarów, to typowe wyzwania w zadaniach inspekcji w takich gałęziach przemysłu jak rolnictwo precyzyjne czy przetwórstwo żywności. Ogólne modele głębokie wykazują niską skuteczność w przetwarzaniu obrazów powstałych w takich warunkach. Ze względu na zło-

żoność tego zagadnienia, to pytanie badawcze zostało rozbite na trzy pytania pomocnicze, które ukierunkowały badania prowadzące do powstania trzech Publikacji (4, 5, oraz 6). Pierwsze pytanie pomocnicze, zaadresowane w Publikacji 4, dotyczyło problemu estymacji wymiarów niewielkich, trójwymiarowych obiektów obrazowanych za pomocą klasycznych kamer bez informacji o głębi. Drugie pytanie pomocnicze polegało na zaproponowaniu metody uczenia modeli *amodalnej* segmentacji instancyjnej, to znaczy modeli zdolnych do segmentacji obiektów częściowo przysłoniętych, bez amodalnej anotacji danych uczących; odpowiednie podejście zaproponowano w Publikacji 5. Trzecie pytanie pomocnicze związane było z poprawą jakości detekcji małych obiektów w obrazach o wysokiej gęstości. W Publikacji 6 zaprezentowana została nowa metoda detekcji obiektów specjalizowana do takich przypadków.

Niniejsza rozprawa podsumowuje rezultaty sześciu pierwszoautorskich publikacji naukowych. W chwili złożenia rozprawy, publikacje te były cytowane 37 razy, co potwierdza ich wpływ naukowy. Ponadto, cztery z tych publikacji zostały opracowane w drodze prac badawczych związanych z realizacją grantów badawczo-rozwojowych fundowanych przez Narodowe Centrum Badań i Rozwoju, co podkreśla ich praktyczną wartość w dziedzinie automatycznej inspekcji wizyjnej.

List of publications

1. **P. Dolata**, M. Mrzygłód, J. Reiner
“*Double-stream Convolutional Neural Networks for Machine Vision Inspection of Natural Products*”
Applied Artificial Intelligence, 31(7–8), pp. 643–659
2017; 20 MEiN points (pre-2018), IF=0.587, 10 citations (WoS)
2. **P. Dolata**, J. Reiner
“*Barley Variety Recognition with Viewpoint-aware Double-stream Convolutional Neural Networks*”
Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, Red., in Annals of Computer Science and Information Systems, vol. 15. IEEE, 2018, pp. 101–105
2018, 70 MEiN points, 7 citations (WoS)
3. **P. Dolata**, J. Reiner
“*Improving Real-Time Performance of U-Nets for Machine Vision in Laser Process Control*”
Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, Red., in Annals of Computer Science and Information Systems, vol. 18. IEEE, 2019, pp. 29–33
2019, 70 MEiN points, Zdzisław Pawlak Best Paper Award
4. **P. Dolata**, P. Wróblewski, M. Mrzygłód, J. Reiner
“*Instance Segmentation of Root Crops and Simulation-based Learning to Estimate their Physical Dimensions for On-line Machine Vision Yield Monitoring*”
Computers and Electronics in Agriculture, vol. 190, pp. 106451
2021, 100 MEiN points, IF=6.757, 20 citations (WoS)
5. **P. Dolata**, P. Majewski, P. Lampa, M. Zięba, J. Reiner
“*Amodal Instance Segmentation for Mealworm Growth Monitoring Using Synthetic Training Images*”
IEEE Access, vol. 13, pp. 52157–52175, 2025
2025, 100 MEiN points, IF=3.6
6. **P. Dolata**, M. Zięba
“*Direct Detection of Elongated Objects Geometry via a Centerline-based Representation*”
Recent Challenges in Intelligent Information and Database Systems. ACIIDS 2025, Nguyen, N.T., et al., in Communications in Computer and Information Science, vol. 2494
2025, 70 MEiN points

1 Introduction

1.1 Background

Industrial automation is the use of machines to improve productivity, quality, and safety across various stages of the processing pipeline. One of the final stages in the processing is quality inspection, which concerns the determination of whether the products conform to some predefined quality standards [41]. The simplest form of quality inspection is to perform a visual inspection by a trained human operator. Industrial inspection can also take the form of process inspection, where the status of some process is monitored, often in real-time. Machine vision aims to automate visual inspection in order to reduce possible errors by the operator and increase efficiency, or allow the inspection in environments where human operation is not feasible [35].

Most generally, a computer vision¹ system comprises four main components: an illumination system, a camera system, computer hardware, and image processing software. The purpose of image processing is to extract information from raw image data in order to support a decision or control system. The classical approach to this task relied on extracting well-defined features from the images and then either constructing a rule-based decision algorithm or building a machine learning model to make predictions given the values of these features. In either case, significant input from a domain expert was necessary to engineer the relevant features. Nonetheless, those systems work very well in applications where identical objects are processed and the inspection can be formulated as the assessment of a deviation from strictly defined norms. Those applications frequently appear in manufacturing industries, most notably in the automotive or semiconductor industries. However, in industries such as agriculture or food processing, objects show a significant intra-class variability, which the classical, rule-driven approach can no longer handle [5].

The rise of machine learning, in particular *deep learning*, offered a radically different, data-driven approach. In the deep learning paradigm, the extraction of manually engineered features is replaced with a deep neural network - typically a convolutional neural network [8] or a transformer [39] - performing *representation learning*, that is, inducing the relevant features (representation) from the training data. Thus, the feature extraction and prediction, previously two distinct steps, are merged into one, large (*deep*) model, which learns end-to-end given a set of training samples and a specifically chosen loss function. Since this approach is no longer restricted to the features a human expert is aware of and a human engineer is able to implement, it has seen successful implementations in innumerable application domains, many of them previously intractable using classical methods [35].

Deep learning comes with its own set of challenges, however. Deep neural networks require a large amount of training data in order to learn meaningful, general features, rather than overfit to some spurious patterns present in the data. In many application scenarios, including industrial inspection, the acquisition and annotation of hundreds of thousands of images for training is infeasible. Such applications follow the transfer learning scheme, in which the problem is matched to some foundational task (such as classification, detection, etc.), then some base model for that task, *pre-trained* on some large, generic dataset, is retrieved, and finally it is *fine-tuned* on the smaller, task-specific dataset. While this process often yields very good results in practice, it is by no means a guarantee, as

¹There is no consensus among scholars on the exact definitions of *machine vision* and *computer vision*. This thesis concerns solely the software aspect of vision, so we shall henceforth use the term *computer vision* without further discussions of the differences.

multiple issues can arise that hinder the performance of a transferred model, or even prevent the transfer outright. Unique conditions of the particular inspection problem might mean that the task does not exactly match the formulation of the fundamental task, for which a pre-trained model exists, necessitating changes in the model itself. Domain-specific data distribution might be sufficiently different from the distribution on which the base model was pre-trained (or even *designed for*), that *negative transfer* occurs - a phenomenon in which the fine-tuned model exhibits degraded performance. Finally, the fine-tuning might be difficult or impossible to perform due to issues related to the training data, such as scarcity, noisy or sparse annotations, or even missing annotations - see [30] for a thorough review.

The application of deep learning in certain industrial inspection scenarios is, therefore, not reducible to a straightforward engineering procedure. In those cases, the development of unique, new methods is required.

1.2 Research scope

The goal of this thesis is to develop specialized deep learning methods for computer vision-based industrial inspection applications in domains where the standard transfer learning approach does not yield satisfying results.

This objective is extremely broad. Therefore, a series of Research Questions was posed in order to more precisely focus the research scope. Below is the list of those Questions, with brief explanations following further in this section. Detailed discussions of the specific issues motivating each related Research Question and the means of addressing them will be given in the respective chapters of this thesis.

- RQ1.** How to construct and train a convolutional neural network-based multi-view classification system for industrial quality inspection?
- RQ2.** How to optimize the architecture of a U-Net-based semantic segmentation model in order to improve its inference time while retaining the segmentation quality?
- RQ3.** How to build computer vision models for the downstream task of dimensioning of small objects in dense scenes from single images?

Research Question 1 concerns the design of image classifiers based on convolutional neural networks for the *multi-view* classification problems in industrial quality inspection. Multi-view classification is a type of image classification task where each sample is given not by a single image, but a tuple of images [40]. This type of problem arises in the food processing industry, where products are imaged from multiple sides in order to, for example, identify defects. Typical convolutional neural networks for classification implicitly assume that a sample *is* the image, which means they cannot be trivially applied to the multi-view task. Chapter 2 breaks this Question further into two sub-questions and discusses publications that answer them. Publication 1, summarized in section 2.2, explores the methods for fusing information from multiple input streams, and Publication 2, summarized in section 2.3, concerns the feature extraction from specialized input streams.

Research Question 2 regards the optimization of a U-Net-based semantic segmentation model to maximize its inference speed with minimal loss of segmentation quality. Image segmentation frequently arises in the industrial inspection context, and the U-Net is a strong baseline model for this task [33]. One such case is laser process monitoring, where the images contain a certain object, whose geometric properties could be used as variables

for real-time process control. The U-Net is capable of segmenting those images with high quality, but does so with a large latency, precluding it from real-time operation. Chapter 3 investigates the reasons for this latency and summarizes the results from Publication 3, which proposes a method of U-Net optimization (section 3.2).

Research Question 3 asks for the methods for building computer vision models for the downstream task of *dimensioning*, specifically for the case of small objects in dense scenes given by a single image. Dimensioning refers to the measurement or estimation of some geometric properties of objects, a problem frequently arising in agriculture or the food industry. In those application domains, it is crucial to process objects on a large scale, often directly on the production line. This leads to images of unstructured scenes containing a possibly large number of relatively small objects, often in highly dense or occluding arrangements. Commonly used deep models for instance segmentation are known to struggle with detection and segmentation of such scenes [10]. In some cases, the task is further complicated by the need to estimate physical dimensions from a single image, without access to depth information. Chapter 4 explores those issues in detail from different angles, posing three additional sub-questions and answering them by detailing results from relevant publications. The first sub-question concerns the problem of size estimation of three-dimensional objects from a single RGB image; section 4.2 summarizes the findings from Publication 4 that addresses this question. The second sub-question asks for the method of training of *amodal* instance segmentation models, that is, models capable of segmenting occluded objects; this question is answered in Publication 5, summarized in section 4.3. The third sub-question is related specifically to the detection of small objects in high-density scenes; results from Publication 6 are discussed in section 4.4 in order to address this question.

The Research Questions stated in this thesis concern important problems preventing the application of deep learning methods for industrial inspection in certain domains. The research undertaken to answer those questions resulted in six publications, all of which I am the first author of. The work on four of those publications was conducted during my employment in research grants funded by the National Centre for Research and Development. Subsequent chapters of this thesis will explain the motivations for each Research Question in greater detail, discuss the relevant conditions of the application domains, and summarize the findings from the publications that address them.

2 Multi-view classification

2.1 Context and research scope

A typical machine learning approach for image classification consists of a feature extraction step, in which a set of features, possibly high-dimensional, is extracted from the input image, and a prediction step, in which a classifier predicts the label for that image [7]. Most of the modern approaches employ neural networks for both steps and train them end-to-end using the cross-entropy loss function. This approach carries an implicit assumption that the input image *is* the sample, considering the problem of sample classification as image classification.

However, this assumption does not hold in certain industrial inspection tasks. In some domains, objects have a non-trivial three-dimensional structure that cannot be completely imaged with a single camera. Features of interest, such as defects, can appear on different sides of the object, necessitating the use of multiple cameras to capture them. As a result, multiple, possibly but not necessarily related, images per instance are acquired. While feature extraction can be performed for each image separately, reasoning has to incorporate information from all input images.

This subtype of image classification, where each sample is given by a pair or a tuple of images, is referred to as *multi-view classification* [40]. General-purpose convolutional neural networks (CNNs) for image classification available in the literature, usually with parameters pre-trained on large datasets, are not well-suited for this kind of task due to the aforementioned assumption of treating sample classification as image classification.

The problem of making a joint prediction for a sample given with two or more images could be formulated in the information fusion framework [24]. The tuple of images representing a sample, although of the same modality, can be considered separate inputs, upon which feature extraction can be performed. Subsequent reasoning must involve the information from all sources; that is, fuse it. Multiple approaches to information fusion are known, such as early fusion, late fusion, and various others, but none of them is universally superior to the others, necessitating a case-by-case investigation.

The available body of knowledge did not provide a definite answer on how to design and train a CNN-based multi-view classification model for industrial inspection. Therefore, **Research Question 1** of this thesis was formulated as follows:

RQ1. How to construct and train a convolutional neural network-based multi-view classification system for industrial quality inspection?

To further narrow down the research, two specific sub-questions were posed and answered in respective Publications, which will be summarized in subsequent sections:

RQ1.1 How to fuse information from multiple images in order to maximize multi-view classification performance?

(Addressed in Publication 1 - see section 2.2)

RQ1.2 How to extract the most valuable features from multiple images in a multi-view classification system?

(Addressed in Publication 2 - see section 2.3)

Both of the studies that this chapter concerns have been undertaken as part of a research grant “Ziarno” funded by the National Centre for Research and Development (grant number PBS3/A8/38/2015). The objective of the grant was to develop a set of methods for

automatic quality assessment of barley grain delivered to a malting house. Barley grains can be plagued with various defects that lower their quality; they can also come in different cultivars (varieties), some of which are more suitable for malting than others [37]. Since the grains have two clearly distinguishable sides, intuition and domain knowledge both suggest that a dual-view imaging and reasoning system could offer superior classification performance. All of the experiments discussed in this chapter were performed on data acquired in the course of this research project.

It might be worth noting that this research was done in the years 2016-2017, when the general state of the art in computer vision-oriented machine learning was not nearly as advanced as in 2025, the year this thesis was compiled. The most recent convolutional neural network backbone for classification at the time was ResNet [14]. Frameworks such as PyTorch [29] and TensorFlow [1] were only seeing their first stable releases (PyTorch in September 2016, TensorFlow in February 2017), and most of the mainstream research, including this, was still done using the (no longer supported) Caffe library [16].

2.2 Multi-view information fusion

This section covers results from Publication 1, which explores the different methods of information fusion for multi-view classification on the example of barley grain quality inspection.

Motivation The base approach for multi-view reasoning adopted in this paper takes inspiration from Siamese networks [2]. In this type of model, two networks using the same set of weights are used to process two different inputs, in order to perform some comparison on them. Typically, such models are trained in the contrastive learning paradigm, using some distance metric.

The same architecture could be used to simultaneously process two images representing the same input, but from different viewpoints. Instead of comparing, the outputs from two sub-networks would be merged to produce a single prediction. A convolutional neural network is a complex sequence of processing layers, however, and there is no obvious point in its architecture to select as a fusion point in a double-stream model. Many different methods of fusion for the purpose of multi-view classification are known in the literature - see [40] for a recent survey. However, there is no evidence for the clear superiority of one approach over the others, especially in the context of industrial inspection. This motivates the question of the exact choice of the information fusion approach for this task.

Content This paper proposes a double-stream convolutional neural network architecture for multi-view classification and compares various methods of combining information inside the network (Figure 2.1). The baseline classification architecture is a CNN based on AlexNet [20], with a reduced final classification layer. In the double-stream model, this convolutional network is duplicated into a *double-stream* network, and the two streams are merged at a certain stage, with the method of merging being a hyperparameter of the model.

Several different methods of information fusion are explored in the experiments. Those include an *early* (or signal-level) fusion, which is equivalent to stitching the two images into a single, larger image, *view-pooling* [36], and different variants of *late* fusion, including averaging [4], prediction-level fusion [34], and a naïve approach in which predictions of two single-image models are averaged. All models are evaluated on an in-house dataset, constructed specifically for the purpose of this study, featuring over 18,000 pairs of images of barley grains in 7 classes (healthy grain vs 6 classes of defects).

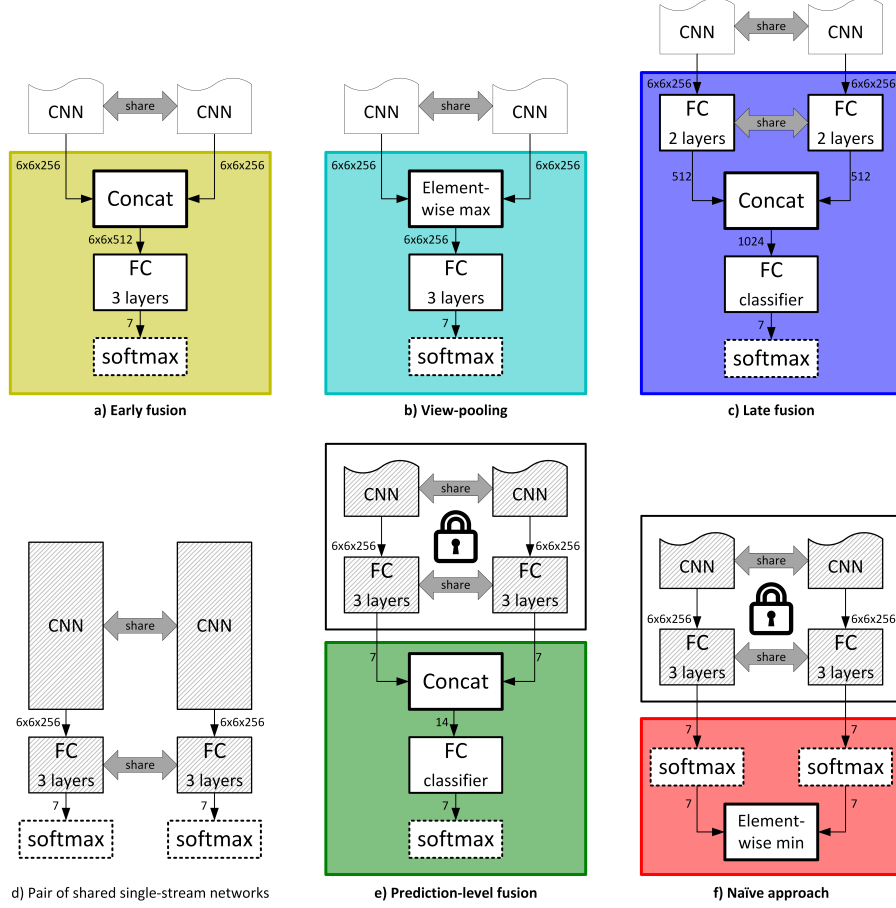


Figure 2.1: The various approaches to information fusion compared in the study. Early fusion (a) merges features extracted by the CNN backbone. View-pooling (b) selects the most active feature from the two representations. Late fusion (c) performs some part of the reasoning on both feature tensors separately and merges the latent representations before making the prediction. Prediction-level fusion (e) simply combines logits from two separate pre-trained classifiers (d), compared to a naïve approach (f) which performs element-wise min operation on their predictions.

The first experiment confirms the validity of the double-stream approach by carefully comparing a selected model against a single-stream baseline, which is found to reduce the false positive rate from 2.21% to 1.81% in binary (“healthy” vs “defect”) classification. Next, different methods of double-stream fusion are compared. Late fusion models turn out to perform better, with prediction-level fusion offering 98.74% accuracy in binary classification and 97.61% in the multi-class task. Surprisingly, depending on the prediction threshold value, the naïve fusion model can perform on par with the strongest models, but evaluation across different values shows that the average area under the curve is significantly higher for the more informed late-fusion models.

Conclusion The paper presented a methodology for designing and training double-stream convolutional neural networks for multi-view classification problems in industrial inspection, demonstrating it on the example of barley grain classification. By comparing several approaches to information fusion in order to classify samples given by multiple images, it addresses Research Question 1.1.

The structure of the paper is such that the methodology is presented in a very general way - it can guide other researchers and practitioners in building multi-view classification architectures dedicated to other domains. It constitutes one of the first demonstrations of this kind in the context of agriculture or food industries.

Contribution I designed all of the neural network architectures (except, naturally, the AlexNet baseline) used in the study, conducted all of the experiments, and analyzed their results; I also wrote the first draft of the manuscript. Mariusz Mrzyglód contributed the details of the comparison methodology, ensuring its rigor, and helped with innumerable technical issues. Jacek Reiner was a guiding force throughout this research, which actually began as my Master’s thesis in 2016, and those results would not have been possible without regular discussions and his feedback, both on the level of experiments as well as the manuscript.

This work was published in Applied Artificial Intelligence (IF=0.587) in 2017 and has received 10 citations according to Web of Science.

2.3 Viewpoint-specific feature extraction

This section covers results from Publication 2, which expands on Publication 1 by introducing a method for multi-view classification of asymmetric objects featuring specialized feature extraction streams.

Motivation In general multi-view classification tasks, such as those described in Publication 1, no assumptions can be made about the order of images representing the sample. For example, a defect of a barley grain can appear on either side of said grain, and the classification model has to extract the relevant feature regardless of its location.

But in other tasks, the appearance of a specific feature on a specific view of the object might have a different consequence. Example involves barley grain variety classification; grains have a clearly distinguishable *dorsal* and *ventral* sides - much like coins, which have heads and tails - and human experts performing variety classification look at the grains from both sides, expecting the presence of certain variety-specific and side-specific traits [37].

This leads to the intuition that an automatic prediction model for this task should be conditioned on features extracted from specific views of the object, as opposed to a viewpoint-agnostic approach of Publication 1.

Content This paper proposes a double-stream CNN architecture in which images of the specific side of the object are deterministically mapped onto specialized feature extraction streams. The convolutional feature extractors are no longer shared - they become *viewpoint-specific*, i.e., one learns features relevant to the dorsal side of the grain, the other to the ventral side. Unfortunately, the imaging system used to acquire the data does not allow for controlling the orientation of the objects, nor would it be possible in practice, should a proposed model be deployed in a real-world quality inspection pipeline. This means that the mapping between object sides and cameras is not readily available and must be estimated for each pair of images based on their content.

For this reason, the proposed architecture consists of two steps (see Figure 2.2, “Oriented” column). First, a smaller CNN model is used to recognize the correct ordering of the images - this step is formalized as a binary classification, where the image pair can be in correct or incorrect order. Then, a specialized double-stream CNN runs on the pair

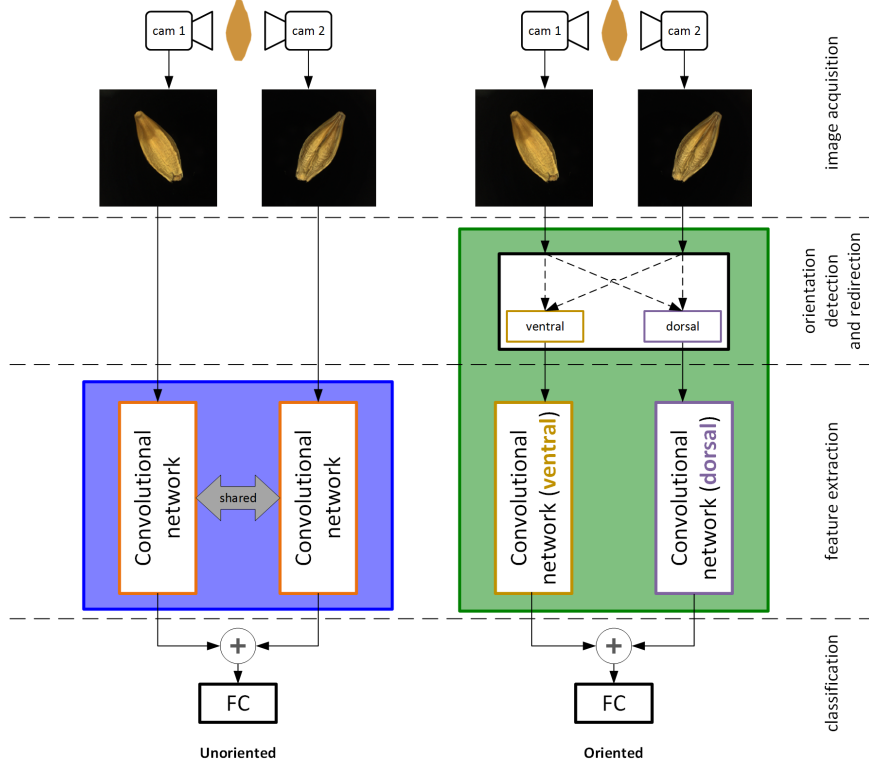


Figure 2.2: The design of a viewpoint-aware double-stream CNN. Compared to the ordinary double-stream network (“unoriented”), which uses the same feature extractor for both input images, the viewpoint-aware model (“oriented”) has two specialized feature extractors. A sub-network for orientation detection and correction is built into the model to facilitate this specialization.

of images in the correct (or *corrected*) order, i.e., images are directed to the specialized feature extractors, and the information from the two streams is combined to form the final prediction. Naturally, it does not matter what *exactly* is the “correct” order, as long as *some* ordering is chosen and maintained throughout the system operation, from training to deployment.

The proposed architecture is not learnable end-to-end due to dependence on the image ordering. Therefore, we first train the ordering network on a separately annotated subset of the training set, and then use this model to annotate the rest of the dataset. Effectively, it is never executed during the training of the viewpoint-aware double-stream classification model, but it would become a part of the final (e.g., deployed) predictive model.

We train the proposed viewpoint-aware double-stream CNN on the challenging task of barley variety classification, on an in-house dataset featuring 3,169 pairs of images in 8 classes. We find that ordering the images and processing them through specialized extractors improves recognition performance. Accuracy improves from 96.65% to 97.24% while sensitivity (averaged across classes) improves from 86.63% to 88.97% for the viewpoint-aware model. However, it is worth noting that this performance boost is not uniform for all classes considered in the research.

Conclusion A novel viewpoint-aware double-stream convolutional neural network architecture was proposed for the task of multi-view classification of asymmetric objects given

by a pair of images. The method is two-stage, with the first stage recognizing the order of images, and the second stage comprising two feature extractors for processing the ordered images. The proposed architecture outperforms the viewpoint-ignorant baseline because the specialized feature extractors make better use of the information available in the input images. This answers Research Question 1.2.

The method not only achieves higher classification accuracy, but also does so in a way more closely resembling the way a human expert approaches the same task. If, for example, the need to explain the decisions of this network had arisen, decoupling the feature extractors would allow the analysis of the specialized streams in separation, possibly yielding more informed explanations.

Contribution I designed the architecture proposed in this paper and conducted all experiments, from implementation to results analysis, with Jacek Reiner advising me throughout all stages. I wrote the initial version of the manuscript, which Jacek then edited.

I presented this work at the 2018 Federated Conference on Computer Science and Information Systems (FedCSIS). To date, it has received 7 citations according to Web of Science.

3 Fast semantic segmentation

3.1 Context and research scope

Semantic segmentation is a computer vision task in which the goal is to assign every pixel a unique class label. Segmentation of an image allows the extraction of more semantic information than a simple classification. There is one significant limitation, however: this paradigm does not allow for distinguishing neighboring entities of the same class. But with sufficient prior knowledge, for example, if only one instance ever appears in the given region of interest, semantic segmentation returns a detailed contour of the object, enabling various modes of downstream processing.

There exists a great number of deep learning-based methods for semantic segmentation - see [12, 23, 38] for attempts at surveying the field. One of the well-known and proven meta-architectures for this task is the U-Net [33]. It achieves very good segmentation quality on various challenging datasets, and does so at a remarkable conceptual simplicity. At the core, semantic segmentation involves two challenges related to feature extraction: high-level features are required to identify complex concepts present in data, but at the same time, low-level features encode boundaries between them. U-Net addresses both of those steps explicitly, connecting low-level but high-resolution information extracted early with highly abstract but low-resolution features. Its simple, generic design, relying exclusively on convolutional, resampling, and activation layers, allowed almost unconstrained application in numerous different domains.

However, in certain industrial inspection applications of semantic segmentation, the quality of segmentation is not the only criterion. In some process control tasks, real-time operation is necessary, and so the low latency of the segmentation model is crucial. Sometimes it might even be acceptable to sacrifice some segmentation quality to improve prediction speed. The standard U-Net architecture, while conceptually simple, involves a number of convolutions and resampling operations performed on the entire input image, leading to long inference times. While this is of little consequence for offline applications such as medical image processing, the real-time operation of this model is impractical.

There are limits to performance improvements that can be made by simply using more powerful computational hardware or using optimization techniques such as quantization. On the other hand, architecture-level optimizations could yield additional improvements, especially when coupled with the aforementioned methods. In the case of industrial process control, some domain knowledge suggests that entire portions of the U-Net model are not doing any important computation. This leads to **Research Question 2** of this thesis:

RQ2. How to optimize the architecture of a U-Net-based semantic segmentation model in order to improve its inference time while retaining the segmentation quality?

This question was addressed in Publication 3, which is summarized in the subsequent section. This publication was a product of research effort undertaken during the realization of a research grant “AMpHOra” funded by the National Centre for Research and Development (grant number INNOLOT/I/6/NCBR/2013). Among the many goals of this grant, one of them involved the development of control methods for laser-based manufacturing processes based on computer vision. By segmenting a certain object appearing on the images acquired during process monitoring, its geometric dimensions could be extracted and utilized as a control variable. However, the dynamics of the process are very rapid, placing tight constraints on the latency of the image processing pipeline [18].

3.2 U-Net architecture optimization

This section covers results from Publication 3, which introduces an efficient, retraining-free method for optimizing a U-Net-based semantic segmentation network architecture.

Motivation The U-Net architecture consists of multiple interleaved convolution and pooling layers organized into two interconnected pathways. The contractive (or “down”) path extracts rich features in multiple levels of detail, incorporating progressively more contextual information, while the expanding (or “up”) path combines the information from multiple levels to produce a high-resolution segmentation map. This allows the extraction of relevant features independently from their level of detail, although at a significant computational cost.

However, given a specific segmentation task, one might find that the relevant features do not appear on all levels of detail uniformly. For example, the target might not depend on high-frequency features significantly; this is intuitively the case with the images acquired during the monitoring of laser processes. If this were the case, a U-Net-based neural network trained on such data would not be able to learn any meaningful patterns *on this level of detail* (while other, e.g., higher, levels still could produce informative representations). This would mean that some portions of the network do not contribute to the final prediction, effectively merely downsampling (and upsampling in the “up” path) the input, albeit at the computational cost of a full convolution.

From this follows the idea that if those layers (or, indeed, entire blocks of the U-Net architecture) could be identified, they could be replaced with a simple resampling operation, reducing the computational cost with little impact on prediction quality.

Content The paper proposes a way to determine which layers can be safely removed from a U-Net-based convolutional neural network. A naïve approach to this problem would be to train multiple versions of a network in different hyperparameter configurations, i.e., with different layers removed, and then evaluate their segmentation performance and computational cost. Performing an exhaustive search of this kind for a model of a given number of layers would require training exponentially many instances, which is prohibitively costly.

Instead, we propose an approach inspired by FractalNet [22], where a model comprises several computational paths that are all merged at inference, but only a random subset of them is active at any iteration of training. Originally, this mechanism was introduced as a means of regularization (hence the name used in the original paper: *drop-path*, intended to relate to dropout). We re-purpose it to train a model so that it becomes robust to having some of its layers disabled (i.e., capable of operating in an incomplete state). As a result, we obtain a single U-Net instance that we can perform post-training surgeries on, that is, deliberately remove specific layers in order to evaluate the performance of the model without them. This is significantly faster than having to retrain a model in every configuration.

We train the U-Net on an in-house dataset of annotated laser process monitoring images. This allows us to perform an ablation study, progressively removing more layers of the network and analyzing the impact of individual layers on the quality of the final prediction (measured using F_1 -score for the classes of interest) and inference time (in milliseconds per image, with no use of batching). We find that, in the case of laser process monitoring images, the early layers of the model are indeed simultaneously the most costly and least important. Removal of the first 3 layers in the “down” path and last 3 in the “up” path

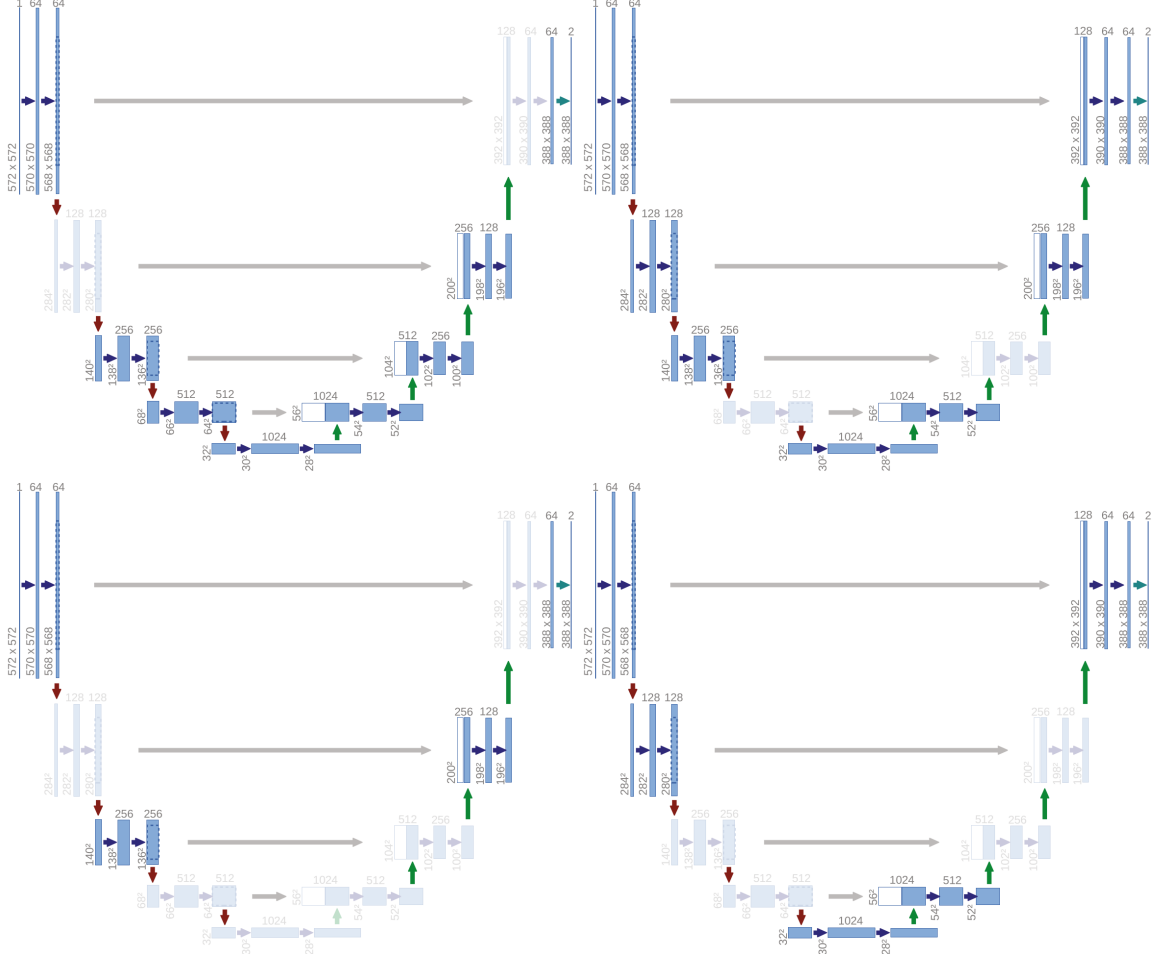


Figure 3.1: The concept of the regularization mechanism proposed to train the U-Net to allow the post-training surgeries. In each iteration of the training loop, a randomly selected subset of the layers is disabled (shown in faded color) and replaced with resampling. This allows the model to learn robustness against surgeries at inference time, allowing the evaluation of layer importance without retraining.

translates to a speed-up of approximately 5x with respect to the original U-Net, while only costing 4 – 11% in F_1 (depending on class).

Conclusion This conference paper introduces a novel method for optimizing (or: *pruning*) a convolutional neural network architecture, with the specific case of U-Nets. The method not only achieves strong results at increasing inference speed while preserving prediction quality, but also does so at a relatively low computational cost itself, thus addressing Research Question 2.

While demonstrated and evaluated on the specific case of laser process monitoring, and exploiting certain properties of the images from this domain, the method itself is generic. Finally, the proposed method does not supersede or prohibit optimization techniques such as quantization - both schemes could be combined for even greater speed-ups.

Nowadays, research like this would most likely be framed in the context of neural network pruning and benefit from the rich body of knowledge in this area. However, at the time the research was done (2018), the state of the literature concerning the pruning of convolutional neural networks was still rather scarce - the two most notable works are Deep Compression [11] and the unpublished Entropy-based Pruning [27]. Without the implementation support, however, comparing our approach against those methods was not feasible. The available computational backends for running CNNs - in this case, the now-abandoned Caffe [16] - did not permit the application of non-CNN-specific pruning techniques, either.

Contribution I came up with the fundamental idea behind this paper, although the final method was a result of many iterations with feedback from Jacek Reiner (and several helpful discussions with Mariusz Mrzygłód). From this point, I designed the network architecture, implemented the training pipeline, and performed all experiments. I also wrote the manuscript, with Jacek aiding with analysis of the results and editing.

I presented this work at the 2019 Federated Conference on Computer Science and Information Systems (FedCSIS) in Leipzig, where it was awarded the Zdzisław Pawlak Best Paper Award.

4 Instance segmentation for object dimensioning

4.1 Context and research scope

Instance segmentation (IS) is a computer vision task that asks to predict segmentation masks (or, equivalently, contours) for each object present in the image. The dominant approach for IS relies on two-stage processing. In a two-stage method, the first stage is object detection, typically formulated as the retrieval of a number (unknown a priori) of bounding boxes corresponding to objects, and the second is binary segmentation within each of the boxes. Depending on the downstream task, the resulting segmentation masks, along with their bounding boxes, can be used for further processing.

Many different models for IS have been published, most notably Mask R-CNN [13] and YOLO (starting with version 5) [17] - see Hafiz et al. [10] or Gu et al. [9] for surveys. The key to the success of those models is the availability of large-scale datasets for training, such as Microsoft COCO [26], PASCAL VOC [6], or Open Images [21]. Training on those large, generic, and well-annotated datasets allowed the models to learn robust and general features, enabling applications in various specific domains via transfer learning. The de facto standard approach for applying IS in practice is to retrieve a pre-trained model from a repository and then fine-tune it on domain-specific data.

This approach often leads to satisfying performance solely by fine-tuning, requiring only a skilled engineer to adjust the training hyperparameters. But certain application scenarios, due to either unique requirements of the complete vision system, or unique limiting constraints, or both, do not lend themselves to the transfer-based treatment. Such scenarios often arise in industrial inspection, particularly quality inspection. Those problems often concern the detection and segmentation of products directly in the production environment, which leads to unique challenges related to image acquisition and annotation. Moreover, the segmentation is rarely the main objective - usually, the contours constitute merely an intermediate representation and are further processed downstream according to the domain-specific logic. To understand the limitations of state-of-the-art instance segmentation methods in the industrial inspection context, we will now expand on the above challenges.

The problem in many quality inspection applications is the measurement or estimation of the physical dimensions of the products. This arises often in agriculture or food processing, and in those contexts, the task is referred to as *dimensioning*. Vision-based dimensioning can be formulated in the IS framework by attaching a downstream processing module to the segmentation model and evaluating the system jointly. This task is complicated significantly when the measured or estimated quantity is related to the three-dimensional structure of the object, but only a single RGB image is available. Only an approximation of the actual quantity can be made in those conditions, for example, via regression. But fitting the regression model might pose a challenge, because acquiring reference measurements in online conditions might be impossible due to logistical constraints.

Practical considerations concerning imaging hardware might necessitate the acquisition of images directly in the existing processing environment. The resulting images, for example, of a running conveyor belt, might contain a large number of small objects. State-of-the-art IS models are known to perform significantly less accurately in those conditions [10]. In the most challenging scenario, particularly in the context of dimensioning, the number of objects can be so large that they occlude one another non-negligibly. Because general IS models fail to segment such instances altogether, dedicated methods were developed with

the explicit purpose of predicting the occluded parts of the objects. This formulation of IS is referred to as *amodal* instance segmentation (AIS) [25]. The training process of AIS models does not differ greatly from the IS models they are often based on, but they require strong supervision. Object-level and pixel-level annotations for IS are already costly to procure, requiring manual labor, often by domain experts if the particular inspection task is complex. The cost of acquisition of amodal annotations is even greater due to the higher manual effort and resulting uncertainty of the annotations [42]. This hinders the application of AIS models in industrial inspection.

Scenes comprising small objects, especially in high-density, occluding arrangements, pose a challenge for IS models. State-of-the-art models are built predominantly with the generic datasets in mind, and those datasets do not contain many images of this difficult type. This leads to a significant difference between the distribution of data used to train the base model and the distribution of the fine-tuning dataset in the industrial inspection task. As identified in preliminary experiments, one of the key failure modes of the IS models is the failure to detect objects, especially in high-density regions. High-density scenes are even more likely to arise in AIS tasks, yet most state-of-the-art AIS approaches focus on improving segmentation performance. The issues of detection rate in those cases remain unaddressed.

As is evident from the above exposition, certain problems of industrial inspection cannot be automated simply by transferring pre-trained state-of-the-art instance segmentation models. This research gap was formulated in **Research Question 3** of this thesis:

RQ3. How to build computer vision models for the downstream task of dimensioning of small objects in dense scenes from single images?

This question encapsulates all of the distinct issues discussed above, making its scope far too broad to be addressed in a single study. For this reason, three sub-questions were formulated in order to guide three separate studies:

RQ3.1 How to estimate the size of three-dimensional objects from a single RGB image?

(Addressed in Publication 4 - see section 4.2)

RQ3.2 How to train amodal instance segmentation models, if no amodal annotations are available?

(Addressed in Publication 5 - see section 4.3)

RQ3.3 How to improve the detection rate of small objects in high-density and high-occlusion scenes?

(Addressed in Publication 6 - see section 4.4)

Publication 4 resulted from research work performed during the realization of a research grant “Biostrateg” funded by the National Centre for Research and Development (grant number BIOSTRATEG3/343547/8/NCBR/2017). The project concerned precision agriculture, with one of its tasks dedicated to yield monitoring and mapping of root crops. This problem involved the 3D size estimation of small objects in very dense arrangements from images acquired via monocular RGB vision, the only practical approach due to the harsh operating conditions of the system and the economic constraints on the commercialization side of the project. The challenge was exacerbated by the lack of downstream-level annotations, which were impossible to acquire, leaving only image-level annotations available.

4.2 3D size estimation from a single image

This section covers results from Publication 4, which concerns the estimation of physical dimensions of three-dimensional objects from single images, demonstrating the approach on the specific use case of root crop yield monitoring.

Motivation The problem of estimating the physical dimensions of an object given an image requires, at a minimum, the segmentation of the object. Note that simple detection may suffice if only crude dimensions of the bounding box are sufficient for the downstream task. For more precise applications, however, the exact boundary of each instance has to be retrieved, necessitating the use of instance segmentation techniques. In order to obtain estimation in some metric coordinates, as opposed to simple pixel counts, camera calibration is necessary; this is particularly important if the imaging system introduces perspective distortion.

If only a single camera is used for image acquisition and, thus, one image per scene is available, information about the three-dimensional structure of the objects is partially lost. This means any reasoning is limited to the two-dimensional plane onto which the objects are projected during imaging. Depending on the application domain, if this projection is a sufficiently precise approximation of the phenomenon, the task of size estimation - or *dimensioning* - can be solved with classical computer vision techniques as a postprocessing step after instance segmentation.

In applications such as root crop yield monitoring, the 3D dimensions of each instance are required, however. In this specific case, it is the *minimum diameter* of each object. The relation between the 3D dimensions of the objects and their 2D projections onto the imaging plane is not well-defined, because the 2D→3D projection is not a bijection, as many 3D objects can have the same projection. For this reason, the task of 3D dimensioning cannot be solved explicitly. Estimations, for example, via nonparametric regression, are a possible approach, but fitting a regression model is impractical due to the logistics of data annotation. While annotating objects in *image space* is relatively simple and can be done offline (an expert simply marks instance boundaries in the image), annotating them in *physical space* requires manual manipulation at the time of data acquisition, which may not be possible. In such a case, the resulting lack of annotation makes training of the size regression model impossible, necessitating the development of a different strategy for this class of problems.

Content This paper introduces a method for building and training a system for estimating 3D dimensions without relying on physical space annotations, outlined in Figure 4.1. The study assumes an approach based on instance segmentation, using an off-the-shelf Mask-RCNN model to detect and segment instances [13], followed by a nonparametric regression model for the downstream task of dimensioning. The regression model is a small feed-forward neural network that takes instance contours as input (constant size representation is obtained by fitting an ellipse) and outputs the desired quantity: the minimum diameter. Image space annotations are available, allowing the training of the instance segmentation backbone, but physical space annotations for the training of the regression model could not be acquired. To work around that issue, the paper proposes to first train the regression model on synthetic data, generated with all necessary annotations, and then to transfer that model onto the domain of real images.

A virtual environment is designed to generate and render images containing instances that match the shape distribution of the physical objects (here, root crops; specifically,

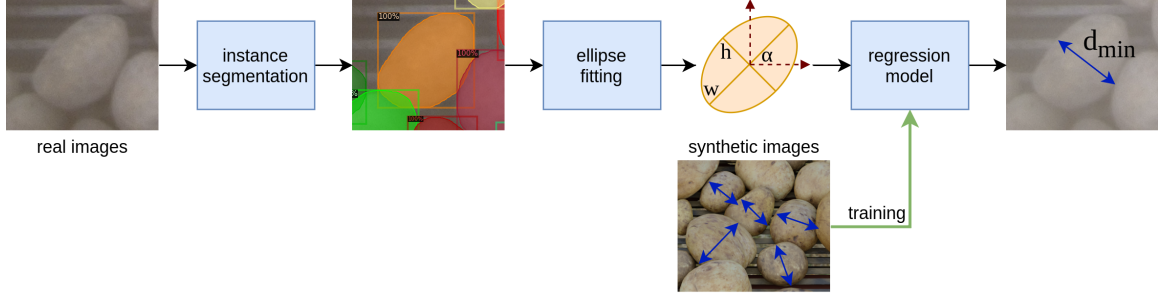


Figure 4.1: Instance segmentation returns contours for each object in the image. Because direct measurement of the desired quantity (minimum diameter) is not possible, a regression model is used to estimate it from an ellipse fit to the contour. This regression model is trained on a synthetic image dataset for which appropriate annotations exist.

potatoes). Scenes rendered this way come fully annotated both on image level, with bounding boxes and contours for each instance, but also on “physical” level, that is, the minimum diameter was also attached to each instance. In order to train the regression model on synthetic data prepared this way, a training set comprising contours matched to true physical dimensions is needed. To obtain it, instance segmentation is performed on the synthetic images first, and the resulting contours are associated with the annotations.

In the proposed setting, the instance segmentation part of the system runs on synthetic images during the training of the regression model, and then on real images during the validation of the final system (and possible deployment). The question arises whether the segmentation masks predicted on the two different domains are of comparable quality, as differences will impact the performance of the downstream model. Instead of transferring the vision model between domains, two separate models are trained: one for the instance segmentation on the real images and the other for the synthetic images. Care is taken to ensure that the models trained on the different domains make similar errors, both in terms of detection and segmentation. As a result, any discrepancies in vision performance are contained in the different instance segmentation models, and the *contours* used for the training and evaluation of the regression model are similar, regardless of the domain of origin. This is the key step that permits the training of a regression model on synthetic images, but evaluating it on real images.

The final system comprises an instance segmentation model - Mask R-CNN - trained on real images, and a minimum diameter regression model trained on contours extracted from synthetic images. The system is evaluated on a completely different test dataset, acquired on a different day and with two different sets of objects. Because, as before, no physical annotations are available for the test set, the evaluation is limited - for example, no RMSE can be calculated on a per-instance basis. Nonetheless, the system is found to distinguish between the two distributions of objects (by qualitative comparison of histograms).

Conclusion This paper proposes an approach for size estimation of three-dimensional objects from single RGB images, tackling specifically the issue of training a regression model without the corresponding annotations, using simulated data instead. To avoid the problem of domain transfer on the instance segmentation level, two separate instance segmentation models are used. Because the errors in the vision domain are matched, the regression model trained on synthetic data can be evaluated on real images as well. This addresses the key

challenge formulated in Research Question 3.1.

The proposed method is not without limitations, but it is not fundamentally restricted to the specific case of root crops, either. The method could be applied in other domains, provided a 3D simulation environment can be constructed for them. This opens possibilities for the application of instance segmentation-based dimensioning in other challenging online scenarios.

Contribution The concept of this study was a joint effort of the entire team, but I proposed the way to train and transfer the regression model. I was responsible for implementing the main processing pipeline, including the instance segmentation model, postprocessing, and the regression model, but with the exception of the synthetic data generation pipeline, which was implemented by Paweł Wróblewski. I also conducted the experiments, analyzed the results, and wrote the original draft of the paper. Mariusz Mrzygłód was responsible for calculations related to the imaging system geometry. Jacek Reiner provided invaluable feedback at all stages of the work and contributed significant edits to the manuscript, ensuring the scientific soundness of the study.

This research was published in *Computers and Electronics in Agriculture* (IF=6.757, 100 MEiN points) in 2021 and, according to Web of Science, has received 20 citations to date.

4.3 Training data for amodal instance segmentation

This section covers results from Publication 5, which introduces a method for preparing data for the training and evaluation of amodal instance segmentation models that requires only modal annotations.

Motivation State-of-the-art models for amodal instance segmentation (AIS) rely on strong supervision for training. Each object in the dataset must have not one, but two corresponding segmentation masks: one in the *modal* domain (i.e., all visible pixels), the other in the *amodal* domain (i.e., all pixels belonging to the object, including the occluded area). In many application domains, there is no shortage of raw data, but the acquisition of the annotations, particularly amodal, is a major challenge.

There have been some attempts to solve the problem of annotated data scarcity by constructing artificial images. The most common avenue involves acquiring fully modal images - that is, containing no occlusions - and then simulating the presence of occlusions by artificially covering patches of the images [19]. This way, the modal annotations, which are relatively easy to acquire, become amodal or partially amodal. However, this comes with a significant limitation: because the occluding patch is either a randomly generated mask or some uninteresting texture (e.g., foliage foreground occluding fruit objects), occlusions between objects of interest cannot be simulated. This approach is entirely ineffective in domains where occlusions arise due to object density, i.e., objects occlude other objects, potentially of the same class.

A competing but less explored direction relies on generative modeling. Here, generative models - based on diffusion models or GANs - are used to generate the training data as well as annotations [15]. Those methods are shown to work well in some applications; however, they incur a very high computational cost, not only at the stage of training of the generative model, but also at inference, as the model has to be queried multiple times to generate a dataset for AIS training. Most importantly, however, this approach does not eliminate

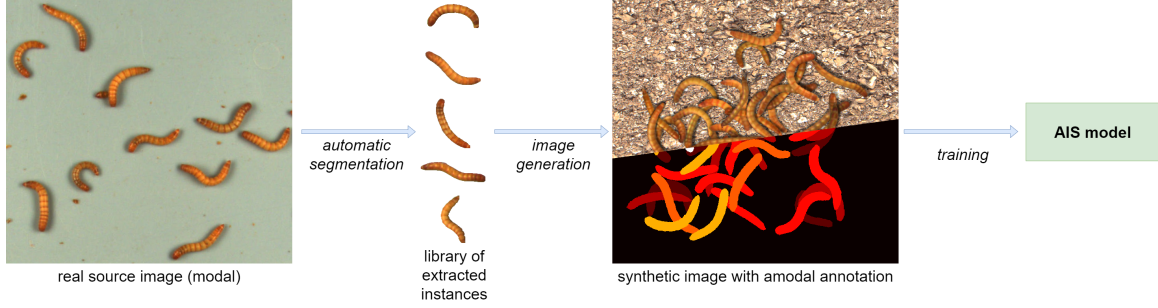


Figure 4.2: Proposed data preparation pipeline. Properly acquired images allow automatic modal segmentation. Instances obtained this way are used to generate arbitrarily dense synthetic images with amodal annotations, which enables the training of an AIS model.

the need to manually create amodal annotations, as some number of labeled examples is still necessary in order to train the generative model. The quantity of available annotated data used to train the generative model will influence the quality of the generated images, impacting the performance of the AIS model trained on them.

In the context of industrial inspection, it might be possible to control the data acquisition process and acquire images with a carefully designed structure that would make them more suitable as a source for the generation of actual training data. The method introduced in this publication is demonstrated on the task of growth monitoring of *Tenebrio molitor* larvae. The inspection problem involves the dimensioning of small objects, and although the imaging conditions are less difficult in this case - the scene is nearly planar, with no need for 3D data acquisition - the scene composition is significantly more challenging. The larvae are small and located in close proximity, frequently occluding one another, even causing some of them to become visually disconnected. However, in the system design phase, the objects can be manipulated at will.

Content This work introduces a novel method for the generation of synthetic images for the training of AIS models. The method does not require the existence of *any* prior amodal annotations, instead assuming that the data acquisition process can be controlled. In outline, the method relies on two steps (see Figure 4.2). First, images are acquired in such a way that objects exhibit negligibly many occlusions, making the instances easy to extract and segment *modally*. Then, amodal images are synthesized by sampling from the extracted modal instances and placing them in a new image in arbitrarily dense arrangements.

In sufficiently controlled imaging conditions, instances can be segmented modally in a fully automatic way even without sophisticated machine learning methods. A collection of entirely visible instances is thus obtained, along with the contours that constitute the annotations; it is worth stressing that both the modal and amodal annotations are identical for entirely visible instances. This enables the key part of the proposed procedure, which is the synthesis of new images that contain occlusions. To do this, a subset of instances is first randomly sampled from the collection. Subsequently, those instances are emplaced in the space of a new image according to an algorithm that allows control over the arrangement of the instances in terms of desired density or occlusion rate. Finally, the RGB images are assembled; the process relies solely on classical computer graphics techniques, making it deterministic, computationally inexpensive, and training-free - only the parameter selection is necessary. Because the object contours are never lost during this sequence of operations,

the complete set of amodal annotations can be assembled in parallel with the images.

Several state-of-the-art AIS models are trained on the generated dataset and then evaluated using AIS-specific metrics (mean average precision) as well as a metric specific to the downstream task (histogram intersection [28]). Importantly, synthetic images are not used for evaluation; instead, a separate dataset comprising only real images is acquired and annotated. (A method that simplifies the process of manual amodal annotation is also given - see section III.B3 in the paper.) Results show that the models perform well on the evaluation dataset, proving the effectiveness of the method. Deeper analysis highlights the impact of object density on detection performance, further motivating Research Question 3.3.

Conclusion This paper proposes a method for the acquisition and generation of training data for amodal instance segmentation models. By exploiting the fact that the image acquisition process can be controlled, the proposed method does not require any manual annotation, modal or amodal, thus addressing Research Question 3.2. Moreover, the actual image generation step is very computationally efficient, especially when compared with generative techniques.

Certain limitations of the method, most notably the controlled data acquisition, mean that it cannot be employed in every conceivable application of AIS. Nonetheless, this study removes a significant obstacle preventing the application of amodal instance segmentation in many practical scenarios where those limitations do not pose an issue. Many instances of industrial inspection tasks fall into this category.

Contribution I initiated this project and organized and led the research group from conceptualization to publication. I proposed the concept for data generation and implemented the synthesis algorithm. I also designed and conducted all experiments and analyzed the results. Finally, I performed the literature review and wrote the bulk of the original manuscript.

Paweł Majewski and Piotr Lampa performed the data acquisition, as well as wrote certain sections of the manuscript. Paweł implemented the phenotyping algorithm used in downstream task evaluation and served the group with his domain knowledge, providing valuable feedback. Piotr implemented the automatic segmentation algorithm that greatly sped up the data preparation process. Maciej Zięba and Jacek Reiner supervised the research methodology and helped with the dissemination of the results and the editing of the manuscript.

This research was published in *IEEE Access* in 2025 (IF=3.6, 100 MEiN points).

4.4 Object detection in high-density images

This section covers results from Publication 6, which follows up on results from Publication 5, addressing the issues related to object detection highlighted therein. Its key achievement is the formulation of the task of object detection using a more geometrically descriptive primitive, a *centerline*, which helps with the detection in high-density images.

Motivation Amodal instance segmentation (AIS) models, which are built as extensions of standard instance segmentation (IS) models, rely on upstream object detection backbones - typically models of Faster R-CNN class [32]. In Publication 5, we confirmed that object detection models struggle with scenes featuring high object density, failing to detect

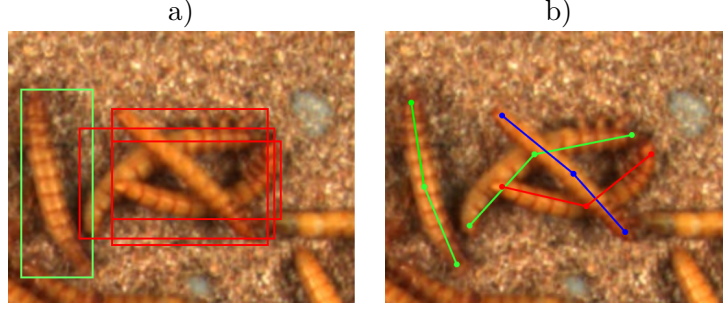


Figure 4.3: In heavily occluded scenes, visually distinct objects can have bounding boxes that are difficult to distinguish (a). A better way to represent the objects might alleviate this problem (b).

overlapping objects. But it is exactly in those high-density regions where occlusions, and, therefore, the need for amodal segmentation, arise in many applications. Thus, improving the detection performance in high-density scenes is crucial for maximizing AIS performance.

The reason why R-CNN-based models fail to detect highly overlapping instances lies in their reliance on Non-Maximum Suppression (NMS). R-CNN-based models, due to their very nature, predict more instances than are actually present in the image; NMS is used to remove excess predictions from the output set. This process relies on the intersection-over-union (IoU) heuristic, assuming that only duplicate predictions will have a high IoU and thus some of them can be pruned. But while this assumption holds in a non-amodal scenario, scenes with high density and significant occlusions - typical in AIS applications - will feature high IoU between correct, ground truth instances, invalidating the heuristic.

Similar issues are present in competing meta-architectures, such as YOLO [31] and DETR [3], but for the sake of brevity, we will focus on the more general underpinning principle instead. All object detection models make one assumption: that *objects* can be uniquely mapped to *bounding boxes* and vice versa. This is indeed the most generic assumption that can be made about any type of object, but in the context of AIS, it is possible for two distinct objects to share an exactly identical box. We hypothesize that making a more domain-specific assumption about the object geometry can reduce the amount of confusion between similar instances, leading to better performance in detection - and, therefore, in amodal instance segmentation.

Content Depending on the specific problem domain, many different assumptions could be made. This study arose in the same application context as Publication 5, mealworm growth monitoring, letting us make a simple observation: all objects in question have a distinctly elongated shape. Following that observation, we propose an alternative way to formulate the object detection task in the specific case of elongated objects. Instead of the generic but non-unique bounding box representation, we shall adopt a more domain-specific geometric primitive: a *centerline* (see Figure 4.3).

The centerline comprises three ordered points that can be interpreted as two connected line segments. It can, therefore, be parameterized with 6 coordinates, as opposed to 4 for a bounding box. Because a centerline provides more geometrical information about the object than a bounding box, the “space of centerlines” is more sparse: it is possible for two objects that occupy the exact same bounding box to have a unique centerline representation. We hypothesize that it is easier to not only detect objects using this representation, but also to

segment them. This is because the centerline identifies an instance more uniquely than a bounding box, providing more informative conditioning input for the segmentation module.

To verify this hypothesis, we reformulate the DETR model to rely on centerlines instead of bounding boxes. We selected DETR as the base model because of its end-to-end, bounding-box-free approach - its outputs are only projected to the box representation in the final layer, as opposed to the R-CNN or YOLO paradigms, which rely on rectangular boxes fundamentally. We modify the output layer of the model and formulate a new loss function for matching and regression of centerlines. We also devise an algorithm for the extraction of centerlines for ground truth annotations from segmentation masks.

The experiments, carried out on simulated data from Publication 5, show improved segmentation results, both in terms of mAP (particularly in high IoU thresholds) but also mAR, which is arguably the more important metric for amodal instance segmentation, where the correct retrieval of objects is challenging. Additionally, the centerline-based approach has one qualitative advantage over the box-based approach - it allows the estimation of certain geometric properties (e.g., length of the object) directly from detection output, without the need for segmentation. We demonstrate this ability on a downstream task of mealworm phenotyping.

Conclusion In this work, we proposed an alternative approach to object detection that improves the detection rate of small objects in high-density and high-occlusion scenes. The method relies on replacing the generic but non-unique representation, the bounding box, with the centerline - a domain-specific representation which offers a unique mapping between objects and representations. This new inductive bias exploits the particular geometry of the objects in the problem domain and, therefore, is not universally applicable. But, with this limitation in mind, this approach addresses Research Question 3.3. Additionally, this method of detection allows the segmentation-free estimation of certain geometric properties, which would not be possible with a bounding box representation.

Although this study presents a particular example of a domain-specific inductive bias for AIS-oriented detection, it proves a more significant result: we can do better than bounding boxes. This highlights a potential for inventing other problem-specific formulations that could bring similar improvements in other domains of application. This hypothesis is an attractive avenue for future work.

Contribution The core idea behind this study, as well as the implementation and experiments, is my work. However, it would not have been possible without the regular discussions and feedback from Maciej Zięba, who also contributed the method for the evaluation of the proposed approach. Additionally, I wrote the original manuscript as well as performed the literature review, with Maciej supporting me with advice and editing.

I presented this research at the poster session at the 17th Asian Conference on Intelligent Information and Database Systems, a CORE B conference (70 MEiN points), in April 2025.

5 Conclusions

This thesis has explored the unique challenges of applying deep learning-based computer vision methods in certain scenarios related to industrial inspection. The current paradigm in applied computer vision relies on general models for fundamental tasks in computer vision, which are pretrained on large datasets and then fine-tuned on domain-specific data via transfer learning. The main assertion of the thesis was that certain application domains feature some unique challenges, which the general models cannot address out of the box, necessitating further research into specialized methods.

A series of research questions was formulated to guide this research, resulting in six publications that address them. The first question concerned the design and training of multi-view classifiers based on convolutional neural networks. Certain classification problems in industrial quality inspection, for example, food products quality control, require reasoning from multiple images corresponding to a single sample, as opposed to the standard classification framework in which an image *is* a sample. Publication 1 introduced double-stream convolutional neural networks for this problem and evaluated several possible strategies for the fusion of information from multiple images. Publication 2 extended those results by proposing viewpoint-aware double-stream CNNs that further improve the classification performance by performing viewpoint recognition and extracting features specialized to a particular view in each stream. Together, those publications describe a methodology for designing multi-view classification networks with a particular focus on quality inspection, addressing the research question. Publications 1 and 2 received a total of 17 citations to date, further confirming the value of this contribution to the scientific community.

The second research question asked for a method for optimizing convolutional neural networks for semantic segmentation, specifically U-Nets. While those models achieve state-of-the-art segmentation quality, their inference speed is too low for real-time applications. Publication 3 addressed this question by proposing a method of training a general U-Net-based model in a way that enables surgeries on the model without subsequent retraining. Via those surgeries, computationally expensive layers that do not substantially contribute to segmentation quality can be removed, increasing inference speed while retaining quality. This publication was awarded the Zdzisław Pawlak Best Paper Award at FedCSIS 2019.

The final research question encapsulated the various challenges of instance segmentation in applications where the downstream task is the size estimation of small objects. State-of-the-art instance segmentation models based on convolutional neural networks and/or transformers, while showing impressive performance on the key benchmark datasets, struggle in certain applications, such as agriculture, for example. The key issues in those domains are the presence of many small objects in high density, the frequent occlusions between the objects, and the need to estimate physical dimensions from a single image. Publication 4 addressed the size estimation problem by proposing a regression model along with a method of training it using simulated data, as the acquisition of ground truth labels in such applications is impractical. Publication 5 proposed a method for training amodal instance segmentation models, which allow the reconstruction of occluded objects but require data annotations that are difficult to obtain in practice. By a careful data acquisition strategy and a novel image generation algorithm, the automatic synthesis of fully annotated training images was made possible, enabling the training of AIS models in domains it was exceptionally difficult before. Publication 6 introduced an alternative formulation of the object detection task, with the goal of improving detection performance of densely arranged and overlapping objects. The new approach blended domain knowledge into the detection

model, proposing a new geometric primitive, a *centerline*, instead of the generally used representation, a bounding box, showing improved detection and segmentation performance.

The research summarized above addressed all of the research questions posed in this thesis. Several areas of applied computer vision have been impacted by the findings. Nonetheless, several directions for future work can be suggested, not only in the area of applications but also in fundamental research.

In semantic and instance segmentation, foundation models are widely predicted to become the approach of choice for most use cases. Certain application domains are less likely to be impacted, however. For reasoning on edge devices, or in highly energy-efficient applications, small and highly specialized models will likely outperform large foundational models. Methods for task-specific optimization or network distillation are a valid research avenue.

Bounding boxes are unlikely to be replaced in the general formulation of the object detection task. Better representations for specific use cases could be studied, however - perhaps with the goal of finding an even more general representation.

The area of amodal instance segmentation is still in its infancy. Issues worth working on include not only the amodal completion itself, but also the detection of overlapping objects. Finally, learning amodal models only from modal labels, perhaps using techniques of weakly supervised learning, would constitute a major breakthrough.

Acknowledgements

While working on this thesis, I was employed at Wrocław University of Science and Technology as a research and teaching assistant. I was also employed as a specialist for the realization of two different grants funded by the National Centre for Research and Development:

- “AMpHOra”, no. INNOLOT/I/6/NCBR/2013
- “Biostrateg”, no. BIOSTRATEG3/343547/8/NCBR/2017

Furthermore, data and computing resources used in the work on Publications 1 and 2 were made available thanks to another NCRD grant, “Ziarno”, no. PBS3/A8/38/2015.

The computing resources used during the study on Publications 5 and 6 were provided by the Department of Artificial Intelligence.

Personal acknowledgements

This thesis is the culmination of many years of my growth as an academic, which would not have been possible without the influence of several important people. I wish to express my gratitude to them here.

Professor Maciej Zięba, for his constant efforts at helping me bring order to the chaos.

Professor Halina Kwaśnicka, for inviting me to the Department of Artificial Intelligence, and for always rooting for me.

Professor Jacek Reiner, without whom I would not have become an academic in the first place.

My parents, for more than this margin can contain.

Notice on the use of generative AI

The entirety of this thesis, including all publications that it summarizes (listed on page 9), has been written by a human. Generative artificial intelligence was not used to generate any portion of the text or the illustrations, with the sole exception of automated grammar checks.

Bibliography

- [1] M. ABADI, A. AGARWAL, P. BARHAM, E. BREVDO, Z. CHEN, C. CITRO, G. S. CORRADO, A. DAVIS, J. DEAN, M. DEVIN, S. GHEMAWAT, I. GOODFELLOW, A. HARP, G. IRVING, M. ISARD, Y. JIA, R. JOZEFOWICZ, L. KAISER, M. KUDLUR, J. LEVENBERG, D. MANÉ, R. MONGA, S. MOORE, D. MURRAY, C. OLAH, M. SCHUSTER, J. SHLENS, B. STEINER, I. SUTSKEVER, K. TALWAR, P. TUCKER, V. VANHOUCHE, V. VASUDEVAN, F. VIÉGAS, O. VINYALS, P. WARDEN, M. WATTENBERG, M. WICKE, Y. YU, AND X. ZHENG, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] J. BROMLEY, I. GUYON, Y. LECUN, E. SÄCKINGER, AND R. SHAH, Signature verification using a siamese time delay neural network, Advances in Neural Information Processing Systems, 6 (1993).
- [3] N. CARION, F. MASSA, G. SYNNAEVE, N. USUNIER, A. KIRILLOV, AND S. ZAGORUYKO, End-to-End Object Detection with Transformers, Springer International Publishing, 2020, p. 213–229.
- [4] D. CIREGAN, U. MEIER, AND J. SCHMIDHUBER, Multi-column deep neural networks for image classification, in 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3642–3649.
- [5] B. DE KETELAERE, N. WOUTERS, I. KALFAS, R. VAN BELLEGHEM, AND W. SAEYS, A fresh look at computer vision for industrial quality control, Quality Engineering, 34 (2022), p. 152–158.
- [6] M. EVERINGHAM, S. M. A. ESLAMI, L. VAN GOOL, C. K. I. WILLIAMS, J. WINN, AND A. ZISSERMAN, The pascal visual object classes challenge: A retrospective, International Journal of Computer Vision, 111 (2015), pp. 98–136.
- [7] R. C. GONZALEZ AND R. E. WOODS, Digital Image Processing, Pearson, Upper Saddle River, NJ, 4 ed., 2017.
- [8] I. GOODFELLOW, Y. BENGIO, AND A. COURVILLE, Deep Learning, MIT Press, 2016. <http://www.deeplearningbook.org>.
- [9] W. GU, S. BAI, AND L. KONG, A review on 2d instance segmentation based on deep neural networks, Image and Vision Computing, 120 (2022), p. 104401.
- [10] A. M. HAFIZ AND G. M. BHAT, A survey on instance segmentation: state of the art, International Journal of Multimedia Information Retrieval, 9 (2020), p. 171–189.
- [11] S. HAN, H. MAO, AND W. J. DALLY, Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, in 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, Y. Bengio and Y. LeCun, eds., 2016.
- [12] S. HAO, Y. ZHOU, AND Y. GUO, A brief survey on semantic segmentation with deep learning, Neurocomputing, 406 (2020), p. 302–321.
- [13] K. HE, G. GKIOXARI, P. DOLLAR, AND R. GIRSHICK, Mask r-cnn, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), Oct 2017.

- [14] K. HE, X. ZHANG, S. REN, AND J. SUN, Deep residual learning for image recognition, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, June 2016, p. 770–778.
- [15] Y. HU, J. WANG, X. WANG, J. YU, AND J. ZHANG, Efficient virtual-to-real dataset synthesis for amodal instance segmentation of occlusion-aware rockfill material gradation detection, *Expert Systems with Applications*, 238 (2024), p. 122046.
- [16] Y. JIA, E. SHELHAMER, J. DONAHUE, S. KARAYEV, J. LONG, R. GIRSHICK, S. GUADARRAMA, AND T. DARRELL, Caffe: Convolutional architecture for fast feature embedding, in Proceedings of the 22nd ACM International Conference on Multimedia, MM '14, ACM, Nov. 2014, p. 675–678.
- [17] G. JOCHER, Yolov5 by ultralytics, 2020.
- [18] P. JUREWICZ, W. RAFAJŁOWICZ, J. REINER, AND E. RAFAJŁOWICZ, Sterowanie procesem napawania laserowego bazujące na przetwarzaniu obrazów, Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2017.
- [19] S. KIM, S.-J. HONG, J. RYU, E. KIM, C.-H. LEE, AND G. KIM, Application of amodal segmentation on cucumber segmentation and occlusion recovery, *Computers and Electronics in Agriculture*, 210 (2023), p. 107847.
- [20] A. KRIZHEVSKY, I. SUTSKEVER, AND G. E. HINTON, Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, 25 (2012).
- [21] A. KUZNETSOVA, H. ROM, N. ALLDRIN, J. UIJLINGS, I. KRASIN, J. PONT-TUSET, S. KAMALI, S. POPOV, M. MALLOCI, A. KOLESNIKOV, T. DUEBIG, AND V. FERRARI, The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale, *International Journal of Computer Vision*, 128 (2020), p. 1956–1981.
- [22] G. LARSSON, M. MAIRE, AND G. SHAKHNAROVICH, Fractalnet: Ultra-deep neural networks without residuals, in International Conference on Learning Representations, 2017.
- [23] F. LATEEF AND Y. RUICHEK, Survey on semantic segmentation using deep learning techniques, *Neurocomputing*, 338 (2019), p. 321–348.
- [24] J. LI, B. ZHANG, AND D. ZHANG, Information Fusion: Machine Learning Methods, Springer Nature Singapore, 2022.
- [25] K. LI AND J. MALIK, Amodal Instance Segmentation, Springer International Publishing, 2016, p. 677–693.
- [26] T.-Y. LIN, M. MAIRE, S. BELONGIE, J. HAYS, P. PERONA, D. RAMANAN, P. DOLLÁR, AND C. L. ZITNICK, Microsoft coco: Common objects in context, in European Conference on Computer Vision, Springer, 2014, pp. 740–755.
- [27] J. LUO AND J. WU, An entropy-based pruning method for CNN compression, *CoRR*, abs/1706.05791 (2017).

- [28] P. MAJEWSKI, P. ZAPOTOCZNY, P. LAMPA, R. BURDUK, AND J. REINER, Multipurpose monitoring system for edible insect breeding based on machine learning, *Scientific Reports*, 12 (2022).
- [29] A. PASZKE, S. GROSS, S. CHINTALA, G. CHANAN, E. YANG, Z. DeVITO, Z. LIN, A. DESMAISON, L. ANTIGA, AND A. LERER, Automatic differentiation in pytorch, in *NIPS-W*, 2017.
- [30] M. PRUNELLA, R. M. SCARDIGNO, D. BUONGIORNO, A. BRUNETTI, N. LONGO, R. CARLI, M. DOTOLI, AND V. BEVILACQUA, Deep learning for automatic vision-based recognition of industrial surface defects: A survey, *IEEE Access*, 11 (2023), p. 43370–43423.
- [31] J. REDMON, S. DIVVALA, R. GIRSHICK, AND A. FARHADI, You only look once: Unified, real-time object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 779–788.
- [32] S. REN, K. HE, R. GIRSHICK, AND J. SUN, Faster r-cnn: Towards real-time object detection with region proposal networks, *Advances in Neural Information Processing Systems*, 28 (2015).
- [33] O. RONNEBERGER, P. FISCHER, AND T. BROX, U-Net: Convolutional Networks for Biomedical Image Segmentation, Springer International Publishing, 2015, p. 234–241.
- [34] G. J. SCOTT, R. A. MARCUM, C. H. DAVIS, AND T. W. NIVIN, Fusion of deep convolutional neural networks for land cover classification of high-resolution imagery, *IEEE Geoscience and Remote Sensing Letters*, 14 (2017), p. 1638–1642.
- [35] M. L. SMITH, L. N. SMITH, AND M. F. HANSEN, The quiet revolution in machine vision - a state-of-the-art survey paper, including historical review, perspectives, and future directions, *Computers in Industry*, 130 (2021), p. 103472.
- [36] H. SU, S. MAJI, E. KALOGERAKIS, AND E. LEARNED-MILLER, Multi-view convolutional neural networks for 3d shape recognition, in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 945–953.
- [37] P. M. SZCZYPÍŃSKI, A. KLEPACZKO, AND P. ZAPOTOCZNY, Identifying barley varieties by computer vision, *Computers and Electronics in Agriculture*, 110 (2015), p. 1–8.
- [38] H. THISANKE, C. DESHAN, K. CHAMITH, S. SENEVIRATNE, R. VIDANAARACHCHI, AND D. HERATH, Semantic segmentation using vision transformers: A survey, *Engineering Applications of Artificial Intelligence*, 126 (2023), p. 106669.
- [39] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, Ł. KAISER, AND I. POLOSUKHIN, Attention is all you need, *Advances in Neural Information Processing Systems*, 30 (2017).
- [40] Z. YU, Z. DONG, C. YU, K. YANG, Z. FAN, AND C. L. P. CHEN, A review on multi-view learning, *Frontiers of Computer Science*, 19 (2024).
- [41] L. ZHOU, L. ZHANG, AND N. KONZ, Computer vision techniques in manufacturing, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53 (2023), p. 105–117.

- [42] Y. ZHU, Y. TIAN, D. METAXAS, AND P. DOLLAR, Semantic amodal segmentation, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

Full texts of publications

The full texts of all publications summarized in this thesis follow after this page, in order:

1. **P. Dolata**, M. Mrzygłód, J. Reiner “*Double-stream Convolutional Neural Networks for Machine Vision Inspection of Natural Products*” Applied Artificial Intelligence, 2017
2. **P. Dolata**, J. Reiner “*Barley Variety Recognition with Viewpoint-aware Double-stream Convolutional Neural Networks*” FedCSIS, 2018
3. **P. Dolata**, J. Reiner “*Improving Real-Time Performance of U-Nets for Machine Vision in Laser Process Control*” FedCSIS, 2019
4. **P. Dolata**, P. Wróblewski, M. Mrzygłód, J. Reiner “*Instance Segmentation of Root Crops and Simulation-based Learning to Estimate their Physical Dimensions for On-line Machine Vision Yield Monitoring*” Computers and Electronics in Agriculture, 2021
5. **P. Dolata**, P. Majewski, P. Lampa, M. Zięba, J. Reiner “*Amodal Instance Segmentation for Mealworm Growth Monitoring Using Synthetic Training Images*” IEEE Access, 2025
6. **P. Dolata**, M. Zięba “*Direct Detection of Elongated Objects Geometry via a Centerline-based Representation*” ACIIDS, 2025



Double-stream Convolutional Neural Networks for Machine Vision Inspection of Natural Products

Przemysław Dolata, Mariusz Mrzygłód, and Jacek Reiner

Centre for Advanced Manufacturing Technologies, Faculty of Mechanical Engineering, Wrocław University of Science and Technology, Wrocław, Poland

ABSTRACT

There are known applications of convolutional neural networks to vision inspection of natural products. For many products it is sufficient to acquire and process a single image, but some might require imaging from two sides. Human experts performing quality inspection of malting barley typically only observe one side of each grain, but in doubtful cases look at both sides, intrinsically combining the information. In this paper, we make two contributions. We present a method for determining whether imaging objects from two sides yields performance benefits over single-sided imaging. Then we introduce a double-stream convolutional network for reasoning from two images simultaneously and analyze several methods of combining information from two streams. We find that when orientation of the object is unpredictable and the streams are not specialized to process a particular view, a fully shared architecture combining information on the prediction level yields best performance (98.7% accuracy on our dataset).

ARTICLE HISTORY

Received 15 September 2017

Accepted 12 January 2018

KEYWORDS

Convolutional neural networks; machine vision; natural products; information fusion

Introduction

Modern food processing industry has to guarantee high quality and safety of products, therefore quality inspection is one of the most important and universal operations. Machine vision methods are widely known as successful tools for automating this inspection in manufacturing. But in the food industry source materials are unique in the fact that they are inherently irregular and variable. They cannot be analyzed with the same algorithms as most synthetic goods. Acceptable quality objects have a high intra-class variance of features, while the inter-class variance between good and defective objects is relatively low, making the task of automating the inspection challenging.

An example of such product is barley grain, a key ingredient in production of beer and whiskey, with high quality requirements. In current industrial practice, identifying defects and contaminations of barley still has to be done visually, based on statistical sampling. A human expert usually looks at multiple grains scattered over a flat surface and hand-picks the defective ones. In case when a

CONTACT Przemysław Dolata ✉ przemyslaw.dolata@pwr.edu.pl 🏢 Faculty of Mechanical Engineering, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/UAAI.

© 2018 Taylor & Francis

decision cannot be made with enough confidence, an individual grain is picked up and viewed with more detail, including rotating to reveal its opposite side. This means that a human is able to perform reasoning based on combined information from multiple views.

Existing approaches to machine vision inspection of samples of barley grain rely on typical document scanners for data acquisition and digital image processing techniques such as morphological operations, edge detection, or segmentation. In these works, arbitrary sets of shape, color, and texture features are extracted from images, so that each grain is represented with a feature vector. Those vectors are then used to train classifiers such as Linear Discriminant Analysis (Zapotoczny, Zielinska, and Nita 2008), artificial neural networks (Nowakowski et al. 2012), hyperellipsoidal decision boundary classifiers (Szczypiński, Klepaczko, and Zapotoczny 2015), or k nearest neighbors (Hailu and Meshesha 2016). However, none of them consider simultaneous processing and reasoning from more than one image of a grain, despite (Szczypiński and Zapotoczny 2012) suggesting that for varietal recognition, both dorsal and ventral sides should be analyzed.

Over the last few years, a new branch of machine learning known as deep learning has improved the state-of-the-art in computer vision. Convolutional neural networks (CNNs) have been particularly successful at tasks such as image classification and object detection. In classical machine learning, a number of arbitrary, hand-engineered features is extracted from image, and then supplied to a trainable classifier. Performance of the model is thus limited by the quality of detected features. The principle behind deep learning is that both classification function and the feature extraction transformation are trained. There is no arbitrariness of a designer (human) and the model learns the most relevant features directly from data.

One of the earliest industrial applications of deep learning was recognition of handwritten postal codes (LeCun et al. 1989). Since then several more complex industrial problems have been solved using CNNs of various architectures—Deng (2014) presents a detailed applications survey. Recent years brought deep learning solutions to natural objects inspection problems (Brahimi, Boukhalfa, and Moussaoui 2017; Grinblat et al. 2016; Lee et al. 2015; Potena, Nardi, and Pretto 2016; Reyes, Caicedo, and Camargo 2015; Sladojevic et al. 2016; Sünderhauf et al. 2014).

The concept of reasoning from multiple images or using multiple classifiers has been researched in the context of deep learning, starting with Cireşan, Meier, and Schmidhuber (2012) who used ensemble methods for road sign recognition. More recently, Scott et al. (2017) proposed a prediction-level fusion of multiple CNN classifiers processing a single image. Other works (Lin, RoyChowdhury, and Maji 2015; Liu et al. 2017), particularly concerning facial recognition (Lu et al. 2017; Xiong et al. 2016), extract different kinds of features and perform classification basing on a fused representation. This

approach has been found successful in processing images of different modalities (Audebert, Saux, and Lefèvre 2017; Su et al. 2015), particularly in video analysis and action recognition (Park et al. 2016). In this area however, prediction-level fusion is also being researched (Li, Chen, and Hu 2017; Simonyan and Zisserman 2014; Ye et al. 2015). Snoek et al. (2005) provide a comparison of early versus late fusion approaches, but their results are not definitely conclusive.

This study is a part of a project to design a complete, automatic barley grain inspection system, capable of imaging individual grains and separating them basing on defects. Whether to equip the system with a single camera or a pair of cameras is an important design choice, influencing costs of the finished device. If imaging the grain from both sides improves classification accuracy, this additional expense will be justified. In this case, the most effective method of combining information from both images needs to be developed.

In this paper, we apply a CNN to barley defect recognition problem, introducing a novel double-stream architecture for simultaneous processing of both sides of a grain. We present our contribution in two distinct sections. In the first, we provide a problem-independent method of examining whether double-sided imaging really increases classification performance of the system. In the second, we analyze several methods of combining information from the dual streams, comparing architectures with different fusion points.

Convolutional neural networks

In computer vision input data are high-dimensional, every pixel of an image constituting an independent dimension. For this reason, classical machine learning algorithms do not input pixel data directly. Instead, images are first processed with feature extraction algorithms in order to reduce representation to a less dimensional form. The difficulty lies in designing this extraction algorithm, so that it provided meaningful information about samples, allowing distinguishing classes in the problem space. CNNs alleviate this difficulty by learning to perform feature extraction.

Connections in those networks are arranged in special patterns, instead of full connectivity between units of subsequent layers. Each unit is only connected to some region of the input space (receptive field). Weights are also constrained, so that units with different receptive fields share the same weights. Effectively, a single layer of a CNN performs a convolution operation on its input, with convolution kernel being a trainable parameter. Stacking convolutional layers with activation functions allows construction of deep networks which learn to detect a hierarchy of features—from simple ones, like edges or color gradients, to more abstract patterns. Those deep CNNs are usually followed by fully connected neural networks (FC) in order to perform classification or regression. This way both the

classification and feature detection functions can be jointly learned directly from data, using known gradient descent algorithms.

There are several unique problems regarding CNNs. The convolution alone does not reduce dimensionality effectively enough—for this reason *pooling* is typically used. Making the networks deep by stacking many layers of neurons is the crucial factor in their performance, but gradients tend to vanish at saturating activation functions. As a solution, a new type of nonlinearity, the rectified linear unit (ReLU), has been introduced (Krizhevsky, Sutskever, and Hinton 2012). Several methods of normalizing responses between multiple convolution kernels have been developed (Ioffe and Szegedy 2015; Krizhevsky, Sutskever, and Hinton 2012). Dropout has been proposed as a countermeasure to overfitting and co-adaptation of units in large networks (Srivastava et al. 2014). Normalization of network outputs into a probability distribution is typically done with the softmax function.

Training a network, assuming each sample belongs to only one of the possible classes, can be understood as minimizing a multinomial logistic loss function:

$$E = -\frac{1}{N} \sum_{i=1}^N \log(p_{i,l})$$

where N is the number of examples in a batch and $p_{i,l}$ is the probability assigned to the correct label for i -th example.

The standard gradient descent is not used to train CNNs: calculating the exact gradient of the loss function is computationally prohibitive due to large number of parameters to optimize and typically large amount of training data. Instead, gradient from only a small portion of the training data (batch) is computed and used to update weights—this is referred to as stochastic gradient descent (SGD).

Experimental setup

Training deep CNNs is still a computationally heavy task and in order to do it efficiently an optimized software implementation is necessary. We chose to use the Caffe framework (Jia et al. 2014), due to its high performance GPU implementation via cuDNN and a freely available repository of pretrained models, the ModelZoo.

Primary CNN architecture used throughout this study is based on AlexNet (Krizhevsky, Sutskever, and Hinton 2012), consisting of five convolutional and three fully connected layers. It was originally designed for the ImageNet classification task (Jia Deng et al. 2009), which is by two orders of magnitude larger in terms of amount of data and number of classes than our problem. Thus we decided to reduce the number of neurons in fully-connected layers from 4096 to

512 neurons to reduce potential of overfitting in the classifier. Additionally, we initialize weights in convolutional layers by transferring them from a model pretrained on ImageNet in order to increase detection performance and limit overfitting in that part of the network.

Double-stream networks

In the case of double-sided imaging, each sample would be given by a pair of images. A naïve solution would be to process each image separately and combine those predictions externally. Xu, Krzyzak, and Suen (1992) give several methods of combining classifiers. But a human expert can view a single grain from many angles during the reasoning process itself, instead of rating two separate images. Intuitively, a network simultaneously processing two images might learn correlations between features found in each of them, potentially providing more accurate predictions at test time.

Since both images are of the same modality (RGB data), they should be processed using the same architecture. Therefore, we decide to duplicate layers from the baseline model in order to create a double-stream network. This network will take a pair of images as input, feeding each one to a corresponding stream. Computation will take place independently for both streams, until the merging point where the information will be combined. The actual choice of this merging point and method of fusion is dependent on the architecture and will be discussed in respective section.

We make no assumptions regarding the arrangement of images—that is, there are no guarantees that, for example, a dorsal side view will always be in the first image of a pair. The network must be robust to randomness in arrangement. We accomplish that by sharing weights and gradients between the duplicated layers, enforcing that their parameters will always be equal. This has the following technical consequence. One training iteration in a single-stream network results in one update of its weights (the following assumes training with simple SGD with momentum):

$$\begin{aligned} V_t &\leftarrow \mu V_{t-1} - \alpha \nabla L(W_{t-1}) \\ W_t &\leftarrow W_{t-1} + V_t \end{aligned}$$

where W_i is the weights vector in iteration i , V_i is the weight updates vector in iteration i , $\nabla L(W)$ is gradient w.r.t. the weights W and μ , α are learning hyperparameters: momentum and learning rate. In a double-stream network, each stream has its own, different gradient $\nabla L^{(1)}$ and $\nabla L^{(2)}$, but the update is still applied to a single weights vector:

$$\begin{aligned} V_t &\leftarrow \mu V_{t-1} - \alpha \nabla L^{(1)}(W_{t-1}) - \alpha \nabla L^{(2)}(W_{t-1}) \\ W_t &\leftarrow W_{t-1} + V_t \end{aligned}$$

This is effectively the same as doubling the batch size for this network, making the training more aggressive. In case of a network that consists of a shared double-stream part merged into a single stream, this makes the shared part learn faster than the single-stream part.

Output of every network will be normalized into a probability distribution using the softmax function. Those distributions will be transformed into correct or false answers differently in two settings: binary classification (grain defective or acceptable) and multi-class classification (index of exact defect or acceptable class). For the multi-class task, probabilities will be converted into indices by choosing the highest ranked element (argmax). For binary classification, the probabilities corresponding to defect classes will be summed and a sample will be marked as acceptable if this collective defect probability will be lower than some threshold. This gives a degree of control over the classifier, allowing tuning of its sensitivity and precision, but is infeasible in cases where no preferred class can be specified (e.g. varietal classification). Additionally, we analyze the binary classifier performance in the case of simply choosing the most probable element (argmax).

Data acquisition and preprocessing

Malting barley grains were obtained from malting house supply and segregated by an expert into seven classes, corresponding to healthy grains and six types of defects and impurities as specified by the Polish Standard (Polish Committee for Standardization 1998).

In order to acquire images of many grains efficiently, a prototype automatic imaging system is used (Lampa, Mrzygłód, and Reiner 2016). Barley is imaged by two color CCD cameras with lenses and strobed LED ring illuminators, installed coaxially above and below a transparent plate (Figure 1). Grains transportation through system is ensured by vibratory and screw feeders and rotation of the plate. Due to additional barriers fixed above the plate each grain is centered in cameras' fields of view. This allows for acquisition of two images per grain: one showing its ventral, another its dorsal side, with efficiency of up to 8 grains per second. However, at such processing speed it is not possible to force the grains to assume any desired orientation in the imaging area—consequently, there is no information which side of the grain is shown in which image.

Images acquired in this process are not modified in any way except for a crop to a square of 800×800 pixels, centered on a grain's visual center of mass, and resize to 227×227 pixels with bilinear interpolation. Directly before putting into the neural network, a mean image of the entire dataset is subtracted from each image, in order to zero-center the data.

We acquired 18,463 pairs of images in seven classes ranging from 160 to 7753 pairs in each class. Figure 2 shows examples of several defects and a good quality grain (bottom most row) for comparison. Data were randomly

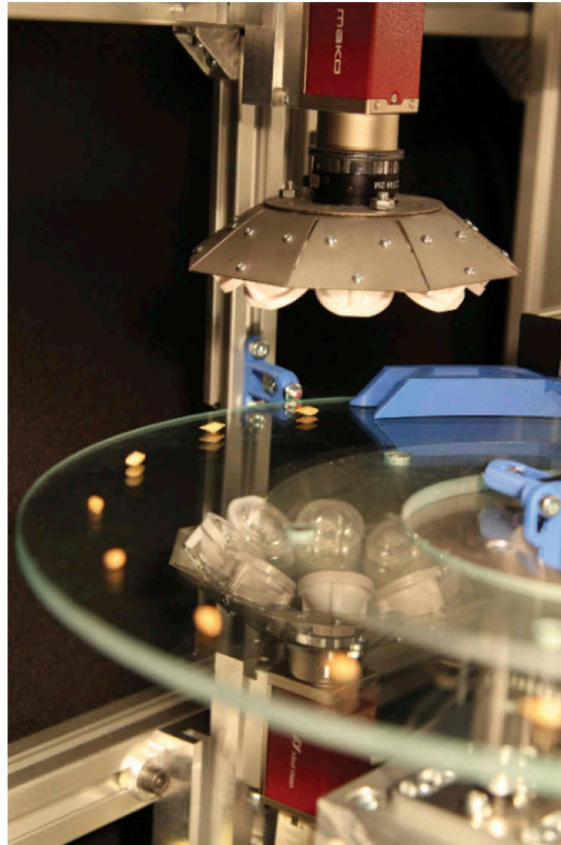


Figure 1. Double-sided image acquisition system.

partitioned into three complementary, approximately equal bins, preserving the class size proportions. These bins were then assembled into three cross-validation folds: each time a different bin was selected as training dataset, with the two remaining ones constituting a validation set.

Comparison of single-stream and double-stream architectures

In this section we demonstrate a method of comparing, whether imaging objects from two sides provides any classification performance benefit over imaging only one side. In the case of double-sided imaging, the resulting dataset would be twice as large as in the latter case, giving the double-sided approach the advantage of more training data. We attempt to isolate the improvement introduced by imaging method itself from the influence of increased dataset size.

For the default double-stream architecture we choose to duplicate the convolutional part of the network, join the extracted features using simple concatenation, and train a fully-connected classifier on that representation. This however introduces the earlier described problem of gradient flow inequality through shared and non-shared parts of the network. In order to provide equal conditions for comparison of single-stream and double-stream architectures, we apply the following procedure ([Figure 3](#)):

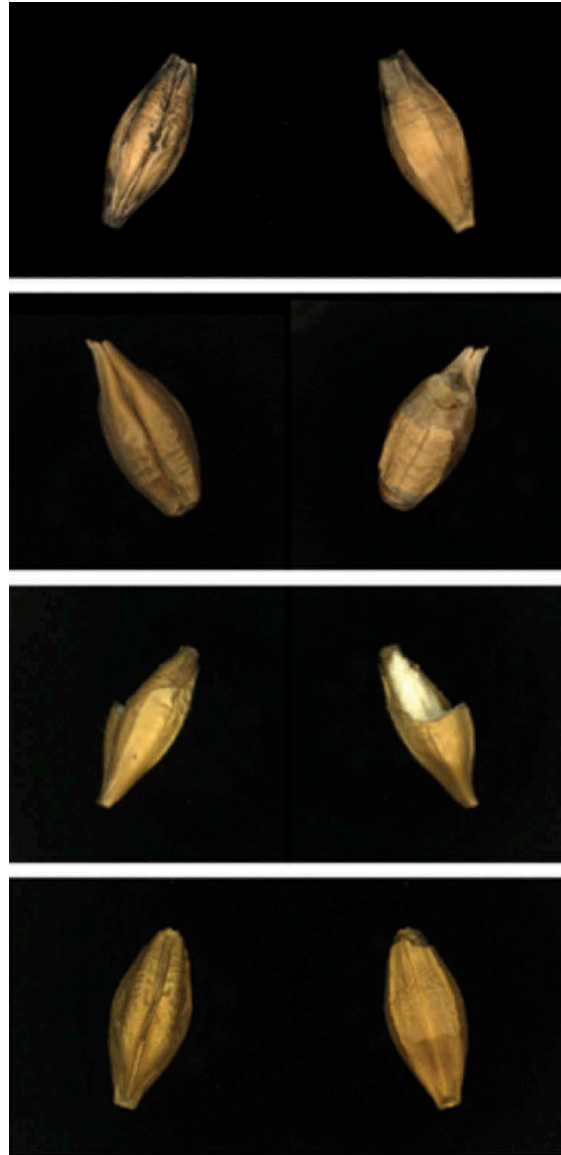


Figure 2. Examples of pairs of acquired images.

- First we train a dummy single-stream network on a dataset consisting of an unordered sequence of images from both cameras (no information about dorsal/ventral correlation is provided at this stage).
- Then we extract weights of its convolutional part and use them to initialize the single-stream and both of the double-stream convolutional parts. Subsequently, we freeze those layers, that is: prevent them from further updates—thus making sure both networks will extract the same features. This will also alleviate the uneven gradient flow in the double-stream architecture.
- Finally we proceed to train the networks, exposing the double-stream model to all pairs of the training set, but picking only one image from each pair for the single-stream network.

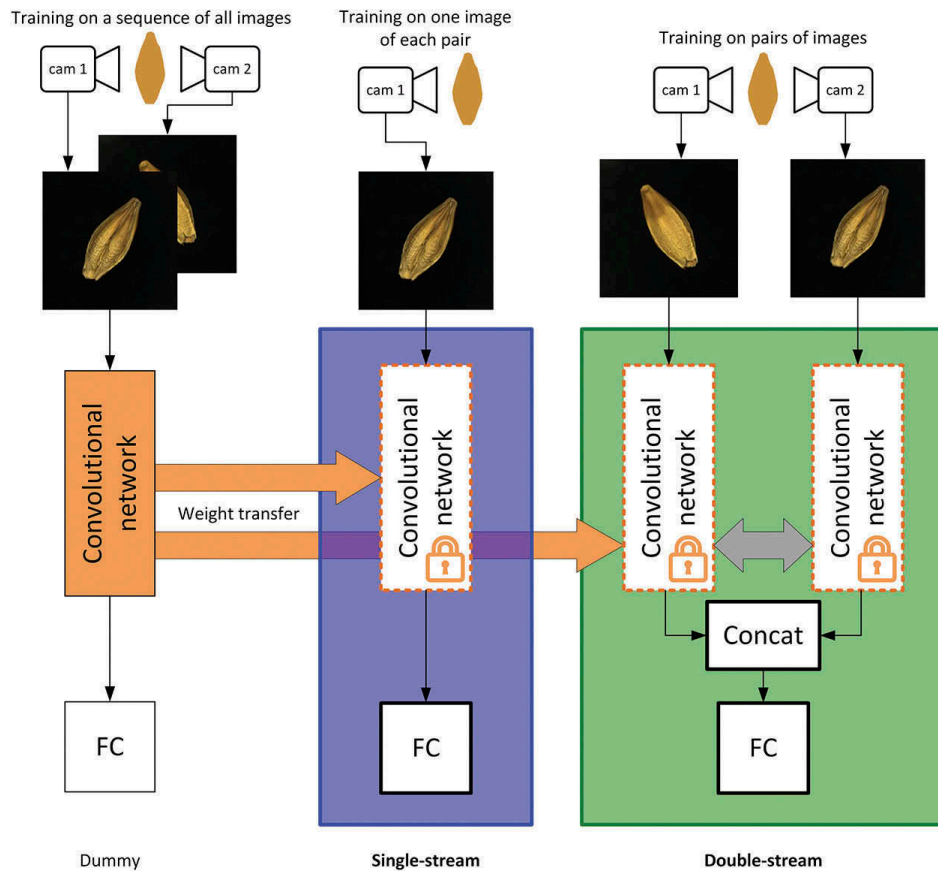


Figure 3. Training procedure.

This ensures equal exposition of feature extractors to training samples, while allowing the classifiers to only learn from the amount of data they would be given during normal operation.

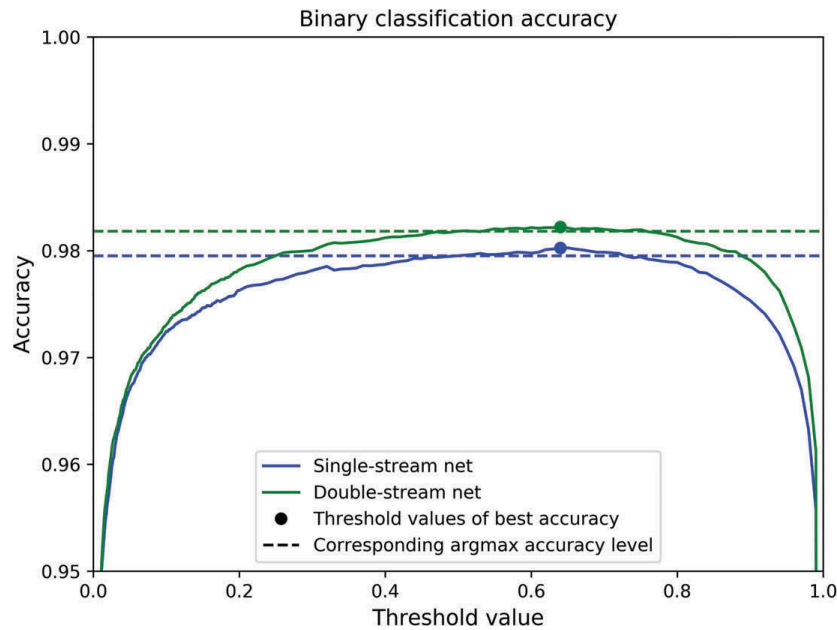
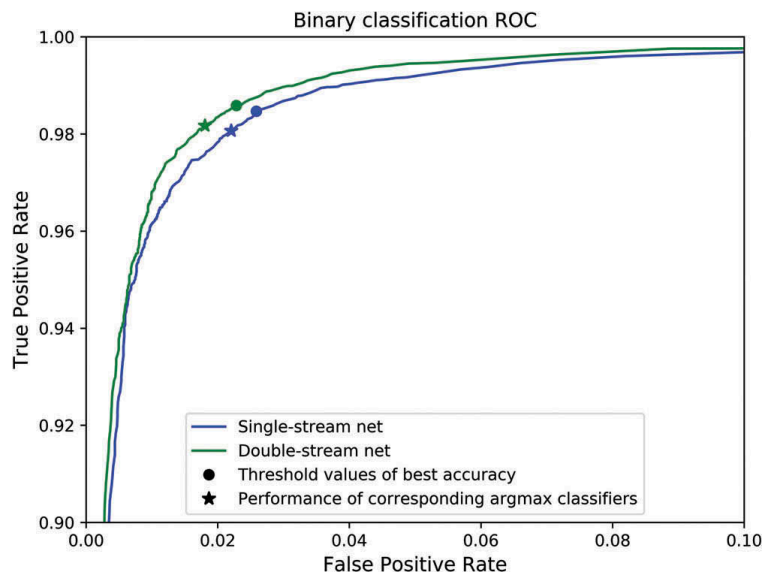
Training hyper-parameters for both networks are the same: we use the standard SGD algorithm with momentum term $\mu = 0.9$, batch size 128 and initial learning rate $\alpha = 0.005$, which is divided by 10 after the first 10 training epochs, and again after the next 3. Results after the total of 15 epochs, averaged between three runs for each model on each of the cross-validation folds, are shown in Table 1.

Figure 4 shows obtained classification accuracy (continuous lines) achieved on the validation set by both models as a function of the threshold value. The maximum accuracy point is marked on each curve. They are additionally compared with performance of the argmax method, shown as a dashed line in the same color as the corresponding curve.

Figure 5 shows the ROC curves for both models, that is plots of false positive ratio on x axis and true positive ratio on the y axis as functions of threshold value, centered on the most interesting region. Performance at the threshold value corresponding to the best binary accuracy is marked with a full circle on each curve. Performance of argmax classifiers is marked with a star. Each point is drawn in the same color as the corresponding curve.

Table 1. Results of the single-stream vs double-stream comparison.

Task	Binary classification				Multi-class classification
	Decision method	Argmax		Threshold (best)	Argmax
Measure	Accuracy	TPR	FPR	Accuracy	Accuracy
Single-stream	97.95%	98.07%	2.21%	98.03%	96.50%
Double-stream	98.18%	98.17%	1.81%	98.22%	96.82%

**Figure 4.** Binary classification accuracy plotted as function of threshold value.**Figure 5.** Binary classification ROC with argmax performance for reference.

Double-stream architecture outperforms the single-stream one by 0.2% in binary classification task, regardless of method of decision-making, and by over 0.3% in multi-class task. Analysis of the ROC proves that the double-stream

yielded better results at every setting of the threshold. The double-stream model made less errors of the first type (false alarms).

It is worth repeating, that results of this experiment will be task-specific, depending on the data. For the case of machine vision inspection of barley grain, we however conclude that imaging objects from both sides will indeed improve classification performance.

Combining information in double-stream convolutional neural networks

Here we present and compare several methods for joining information from two convolutional network streams. We begin with the early fusion model (Figure 6a) and the view-pooling method of Su et al. (2015) (Figure 6b). In both of them the convolutional parts are shared and their outputs are joined before the fully connected classifier. In the early fusion model this is realized by simple concatenation and in the view-pooling method by an element-wise max operation (Eq. 1).

$$y_i = \max(a_i, b_i) \quad (1)$$

In the next architecture, the late fusion (Figure 6c), streams are not joined until the final classifier layer. The convolutional parts and first two of the fully-connected layers are shared, their outputs are concatenated and this intermediate vector is an input to the final layer.

In this experiment, it is assumed that all models will be trained and deployed on pairs of images, so there is no need for special conditions ensuring equal comparison. Therefore, those three models can be learned jointly in a uniform setting, the same as in the previous experiment.

This is not the case with the prediction-level fusion. In this architecture the entire networks are shared, including the classifiers which are however stripped of the softmax function. Raw fully-connected outputs are concatenated and input to an additional classifying layer. This model is trained in two stages: first the shared streams are trained as single networks without any fusion (Figure 6d), then they are frozen and the additional classifier is trained on their outputs (Figure 6e).

These architectures are finally compared to the naïve approach. This is to train only one network like in the first stage of the prediction-level fusion model (Figure 6d), and test it on two images separately, performing the fusion of predictions for each image externally (Figure 6f). This operation can be performed in several ways, depending on the decision mode (e.g. request that both streams assign at least some minimum confidence to the preferred class, average the distributions). We have found that simply calculating the average of two outputs yields the best performance, therefore we chose to use this method in comparisons.

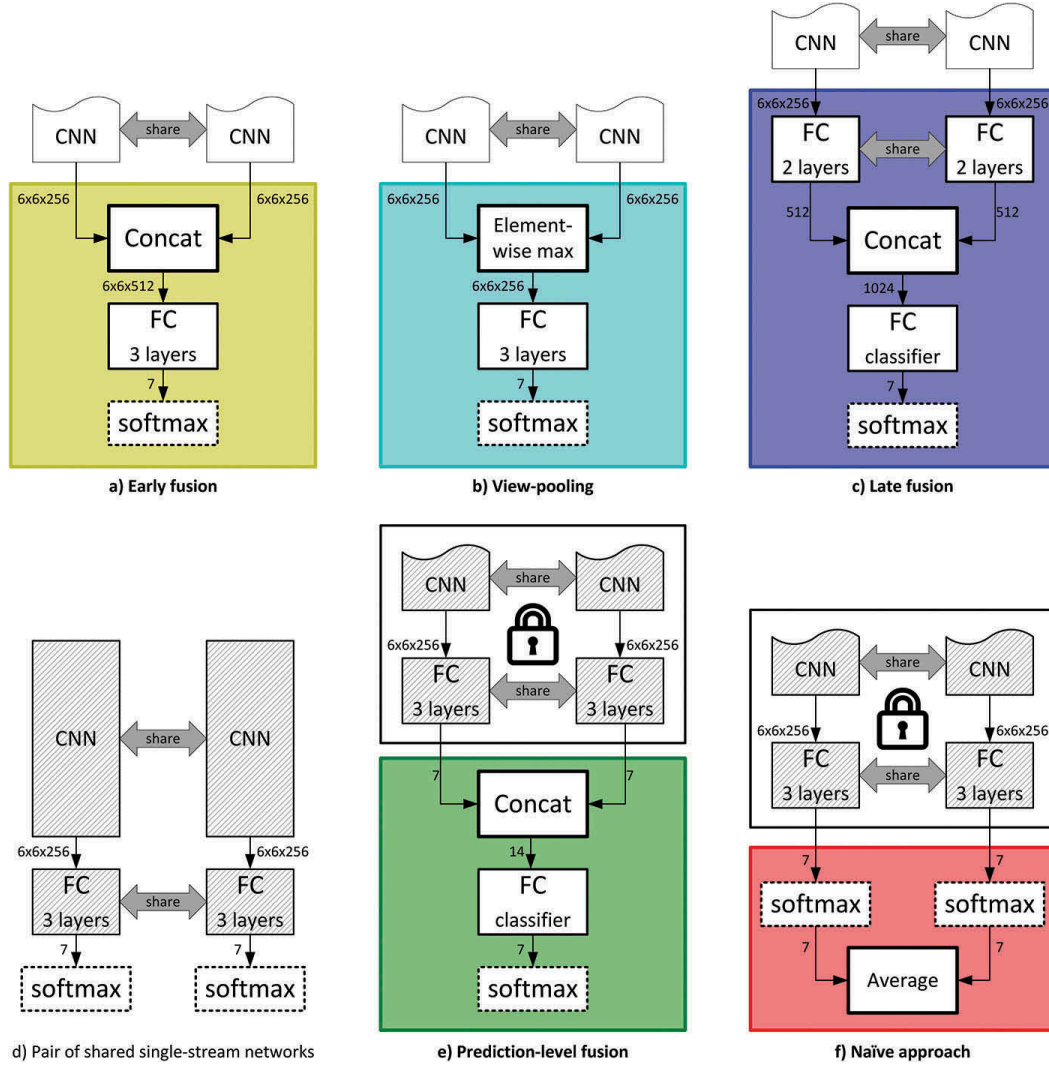


Figure 6. Fusion architectures.

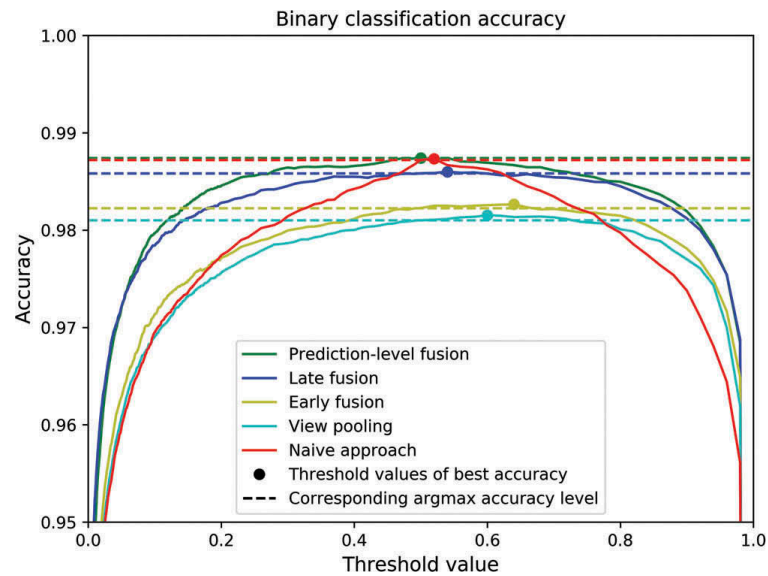
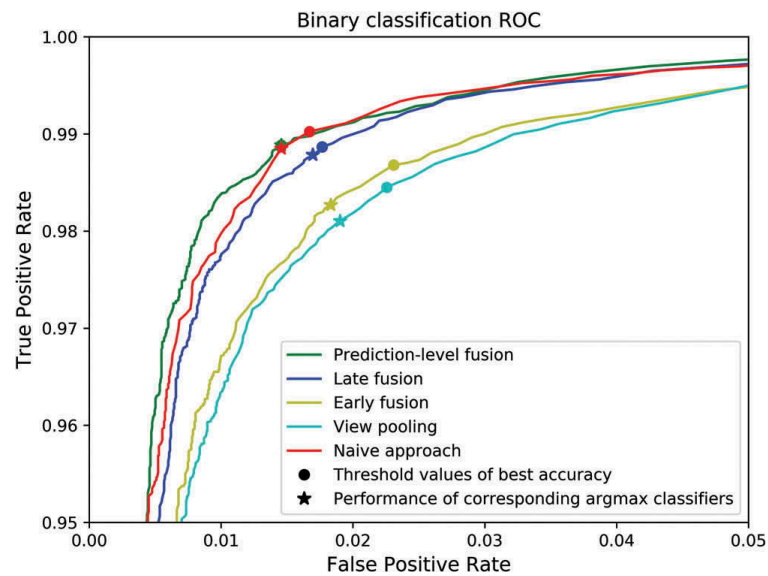
Table 2 summarizes performance of the trained models, as tested on respective validation sets and averaged over three cross-validation folds. Figure 7 shows a plot of binary accuracy versus threshold setting along with argmax performance while Figure 8 compares ROC. Both figures are organized as in the previous experiment.

Architectures with late fusion perform generally better than those with early fusion. Models with convolutional feature fusion are significantly less accurate than those where merging occurs near or after the classifier, as much as 0.6% less accuracy in binary classification and 1.5% less in the multi-class task. The view-pooling method yielded worst results. We suppose this is due to the element-wise max operation only propagating gradients to the stream which produced the maximum output. This means that gradients are split between streams instead of shared, effectively slowing down the learning for that part of the model.

Surprisingly, the naïve classifier performs on par with the best, prediction-level model, equaling its performance in binary task and surpassing it by 0.1% in

Table 2. Results of comparison of various fusion models, best results in bold.

Task	Binary classification			Multi-class classification	
	Argmax			Threshold	
Decision method	Accuracy	TPR	FPR	Accuracy	Accuracy
Early fusion	98.23%	98.27%	1.83%	98.26%	96.26%
View pooling	98.10%	98.10%	1.90%	98.15%	96.08%
Late fusion	98.58%	98.79%	1.70%	98.60%	97.28%
Prediction-level f.	98.74%	98.88%	1.46%	98.74%	97.61%
Naïve approach	98.72%	98.85%	1.46%	98.73%	97.70%

**Figure 7.** Binary classification accuracy plotted as function of threshold value.**Figure 8.** Binary classification ROC with argmax performance for reference.

multi-class. This might mean that none of the double-stream architectures were able to learn any conditional relationships between features extracted from two images. It could be due to the fact that images are fed into the networks with no

predictable arrangement—results would have been different if either the grains could be forced to assume a specific orientation in the imaging device, or this orientation could be recognized in a preprocessing step. However, in an industrial environment requiring high processing speed, this might not be feasible.

Accuracy plots in binary classification with threshold decision mode reveal the primary weakness of the naïve approach. While the double-stream network with prediction-level fusion remains highly accurate over a much wider range of values, the naïve classifier only has a single accuracy peak—its performance quickly degrades in both directions. This means that the double-stream architecture gives more freedom in setting the cut-off point and allows more customization of balance between sensitivity and precision of the classifier.

Conclusions

We introduced a double-stream CNN architecture for double-sided machine vision inspection of natural products on the example of barley grains. We presented a method of verifying, whether such approach yields classification performance benefit, separating the improved architecture factor from the influence of simply increased amount of training data. For the task of barley classification, it proved that double-sided imaging results in better performance (98.22% vs 98.03% for single-sided approach).

We analyzed several information fusion models, testing them on the task of barley grain classification. We found that the best double-stream architecture with prediction-level fusion performs almost equally well as a naïve approach of averaging two independent predictions. Both models achieved 98.74% accuracy in binary and, respectively, 97.61% and 97.70% in multi-class classification. Still, the double-stream network, due to a more stable accuracy characteristic, allows a greater freedom of tuning the binary classifier.

The reason that the most straightforward approach was as effective as the complex double-stream network might have been the fact that images were assigned to the streams in a random order. Each side of the grain looks differently, potentially containing different features to extract. An alternative approach would be to use a model consisting of two specialized networks to process dorsal and ventral sides separately, together with a preprocessing step to identify the orientation of a grain. This might provide performance improvement over a model processing both images agnostic of their arrangement.

Acknowledgements

We wish to thank Piotr Lampa for performing data acquisition. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. However, the source material (barley grain) for the dataset preparation was supplied from a project financed by National Centre for Research and Development—Project PBS3/A8/38/2015.

Funding

National Centre for Research and Development—Project PBS3/A8/38/2015

References

- Audebert, N., B. Le Saux, and S. Lefèvre, 2017. Fusion of heterogeneous data in convolutional networks for urban semantic labeling. Presented at 2017 Joint Urban Remote Sensing Event (JURSE), Dubai, UAE, pp. 1–4. doi: [10.1109/JURSE.2017.7924566](https://doi.org/10.1109/JURSE.2017.7924566)
- Brahimi, M., K. Boukhalfa, and A. Moussaoui. 2017. Fusion of heterogeneous data in convolutional networks for urban semantic labeling. Presented at 2017 Joint Urban Remote Sensing Event (JURSE), Dubai, UAE, pp. 1–4. doi: [10.1109/JURSE.2017.7924566](https://doi.org/10.1109/JURSE.2017.7924566).
- Cireşan, D., U. Meier, and J. Schmidhuber. 2012. Multi-column deep neural networks for image classification. Presented at 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, pp. 3642–3649. doi: [10.1109/CVPR.2012.6248110](https://doi.org/10.1109/CVPR.2012.6248110).
- Deng J., W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. Presented at 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, pp. 248–255. doi: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848)
- Deng, L. 2014. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing*, 3. doi: [10.1017/atsip.2013.9](https://doi.org/10.1017/atsip.2013.9)
- Grinblat, G. L., L. C. Uzal, M. G. Larese, and P. M. Granitto. 2016. Deep learning for plant identification using vein morphological patterns. *Computers and Electronics in Agriculture* 127:418–24. doi: [10.1016/j.compag.2016.07.003](https://doi.org/10.1016/j.compag.2016.07.003).
- Hailu, B., and M. Meshesha, 2016. Applying image processing for malt-barley seed identification. Presented at the Conference: Ethiopian the 9th ICT Annual Conference 2016 (EICTAC 2016), Addis Ababa.
- Ioffe, S., and C. Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning*, 37:448–456
- Jia, Y., E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *Proceedings of the 22nd ACM international conference on Multimedia (MM '14)*. ACM, New York, NY, USA, 675–678. doi: [10.1145/2647868.2654889](https://doi.org/10.1145/2647868.2654889)
- Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems* 25:1097–1105
- Lampa, P., M. Mrzygłód, and J. Reiner. 2016. Methods of manipulation and image acquisition of natural products on the example of cereal grains. *Control & Cybernetics* 45 (3):339–354
- LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation* 1 (4):541–51. doi: [10.1162/neco.1989.1.4.541](https://doi.org/10.1162/neco.1989.1.4.541).
- Lee, S. H., C. S. Chan, P. Wilkin, and P. Remagnino, 2015. Deep-plant: Plant identification with convolutional neural networks, in: 2015 IEEE international conference on image processing (ICIP). Presented at the 2015 IEEE International Conference on Image Processing (ICIP), pp. 452–56. doi: [10.1109/ICIP.2015.7350839](https://doi.org/10.1109/ICIP.2015.7350839)
- Li, H., J. Chen, and R. Hu. 2017. Multiple feature fusion in convolutional neural networks for action recognition. *Wuhan Univ. J. Nat. Sci.* 22:73–78. doi: [10.1007/s11859-017-1219-4](https://doi.org/10.1007/s11859-017-1219-4).

- Lin, T.-Y., A. RoyChowdhury, and S. Maji, 2015. Bilinear CNNs for fine-grained visual recognition, in: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 1449–57.
- Liu, Y., X. Chen, H. Peng, and Z. Wang. 2017. Multi-focus image fusion with a deep convolutional neural network. *Information Fusion* 36:191–207. doi:[10.1016/j.inffus.2016.12.001](https://doi.org/10.1016/j.inffus.2016.12.001).
- Lu, X., X. Duan, X. Mao, Y. Li, and X. Zhang. 2017. Feature extraction and fusion using deep convolutional neural networks for face detection. *Mathematical Problems in Engineering* 2017:1–9. doi:[10.1155/2017/1376726](https://doi.org/10.1155/2017/1376726).
- Nowakowski, K., P. Boniecki, R. J. Tomczak, S. Kujawa, and B. Raba, 2012. Identification of malting barley varieties using computer image analysis and artificial neural networks. Presented at the Fourth International Conference on Digital Image Processing (ICDIP 2012), International Society for Optics and Photonics, p. 833425. doi:[10.1117/12.954155](https://doi.org/10.1117/12.954155)
- Park, E., X. Han, T. L. Berg, and A. C. Berg, 2016. Combining multiple sources of knowledge in deep CNNs for action recognition, in: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). Presented at the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–8. doi:[10.1109/WACV.2016.7477589](https://doi.org/10.1109/WACV.2016.7477589)
- Polish Committee for Standardization, 1998. PN-R-74110:1998P. Barley – Test methods.
- Potena, C., D. Nardi, and A. Pretto, 2016. Fast and accurate crop and weed identification with summarized train sets for precision agriculture, in: *Intelligent Autonomous Systems 14*. Presented at the International Conference on Intelligent Autonomous Systems, Springer, Cham, pp. 105–21. doi:[10.1007/978-3-319-48036-7_9](https://doi.org/10.1007/978-3-319-48036-7_9)
- Reyes, A. K., J. C. Caicedo, and J. E. Camargo, 2015. Fine-tuning deep convolutional networks for plant recognition, in: *Working Notes of CLEF 2015 Conference*.
- Scott, G. J., R. A. Marcum, C. H. Davis and T. W. Nivin. 2017. Fusion of Deep Convolutional Neural Networks for Land Cover Classification of High-Resolution Imagery. *IEEE Geoscience and Remote Sensing Letters* 14 (9):1638–1642. doi: [10.1109/LGRS.2017.2722988](https://doi.org/10.1109/LGRS.2017.2722988)
- Simonyan, K., and A. Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. *Advances in Neural Information Processing Systems* 27:568–576
- Sladojevic, S., M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic. 2016. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience* 2016:1–11. doi:[10.1155/2016/3289801](https://doi.org/10.1155/2016/3289801).
- Snoek, C. G. M., M. Worring, A. W. M. Smeulders. 2005. Early Versus Late Fusion in Semantic Video Analysis. In: *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pp.399–402. doi: [10.1145/1101149.1101236](https://doi.org/10.1145/1101149.1101236)
- Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–58.
- Su, H., S. Maji, E. Kalogerakis, and E. Learned-Miller, 2015. Multi-view convolutional neural networks for 3d shape recognition. Presented at the *Proceedings of the IEEE International Conference on Computer Vision*, pp. 945–53.
- Sünderhauf, N., C. McCool, B. Upcroft, and T. Perez. 2014. Fine-grained Plant Classification Using Convolutional Neural Networks for Feature Extraction, In *Working Notes of CLEF 2014 Conference*, Eds. L. Cappellato, N. Ferro, M. Halvey, and W. Kraaij, pp. 756–762. Sheffield, The United Kingdom.
- Szczypiński, P. M., A. Klepaczko, and P. Zapotoczny. 2015. Identifying barley varieties by computer vision. *Computers and Electronics in Agriculture* 110:1–8. doi:[10.1016/j.compag.2014.09.016](https://doi.org/10.1016/j.compag.2014.09.016).

- Szczypiński, P. M., and P. Zapotoczny. 2012. Computer vision algorithm for barley kernel identification, orientation estimation and surface structure assessment. *Computers and Electronics in Agriculture* 87:32–38. doi:[10.1016/j.compag.2012.05.014](https://doi.org/10.1016/j.compag.2012.05.014).
- Xiong, C., L. Liu, X. Zhao, S. Yan, and T. K. Kim. 2016. Convolutional fusion network for face verification in the wild. *IEEE Transactions on Circuits and Systems for Video Technology* 26 (3):517–528. doi:[10.1109/TCSVT.2015.2406191](https://doi.org/10.1109/TCSVT.2015.2406191).
- Xu, L., A. Krzyzak, and C. Y. Suen. 1992. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics* 22 (3):418–435. doi:[10.1109/21.155943](https://doi.org/10.1109/21.155943).
- Ye, H., Z. Wu, R.-W. Zhao, X. Wang, Y.-G. Jiang, and X. Xue, 2015. Evaluating two-stream cnn for video classification, in: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. ACM, pp. 435–42. doi:[10.1145/2671188.2749406](https://doi.org/10.1145/2671188.2749406).
- Zapotoczny, P., M. Zielinska, and Z. Nita. 2008. Application of image analysis for the varietal classification of barley. *Journal of Cereal Science* 48 (1):104–110. doi:[10.1016/j.jcs.2007.08.006](https://doi.org/10.1016/j.jcs.2007.08.006).

Barley Variety Recognition with Viewpoint-aware Double-stream Convolutional Neural Networks

Przemysław Dolata

Wrocław University of Science and Technology
ul. Wyb. Wyspiańskiego 27,
50-370 Wrocław, Poland
Email: przemyslaw.dolata@pwr.edu.pl

Jacek Reiner

Wrocław University of Science and Technology
ul. Wyb. Wyspiańskiego 27,
50-370 Wrocław, Poland
Email: jacek.reiner@pwr.edu.pl

Abstract—Varietal homogeneity is an important factor in quality of malting barley, but its inspection is difficult. Biochemical methods are expensive and inefficient, while machine vision suffers due to high variability of the grains' features. In our previous work, we have shown a convolutional neural network for simultaneous feature extraction and classification of image data basing on multiple views. It was suggested that for machine vision inspection, the observed side of a grain should be taken into account – dorsal and ventral sides of each kernel exhibit different features. In this study we present a viewpoint-aware convolutional neural network, which learns to extract specialized features from images of dorsal and ventral sides of barley grains. We show that it increases the average classification accuracy by 0.6% and sensitivity by 2.3% with respect to the viewpoint-ignorant architecture on our dataset.

I. INTRODUCTION

In the case of food products, the quality of ingredients which they are produced from plays a significant role, hence their comprehensive inspection is necessary. The sooner the potential fault can be detected, the lower is the actual production cost including wasted materials. Therefore, effective and quick quality assessment requires automation. Among non-invasive methods, machine vision has a special significance. However, the difficulty in identifying quantitative features and their considerable variability recently draws attention to artificial intelligence methods, especially deep learning.

An example of such a natural product is barley, especially its malting varieties, which is a key ingredient in the production of beer and whiskey. Any deficiencies in its quality immediately affect quality and therefore value of the finished product. Hence, the examination of purchased barley includes detection of impurities or damage as well as moisture content and protein content measurement. Typically, such assessment is performed visually by sampled statistical process control (SPC) as it is technically infeasible to inspect all individual grains in a shipment weighing several tons. This process is however very tedious and, due to the difficulty of the task as well as the human's fatigue, error-prone.

This work was supported from the internal budget of Mechanical Faculty of WUST. The source material (barley grain) for the dataset preparation was supplied from a project financed by National Centre for Research and Development - Project PBS3/A8/38/2015.

The subtle flavors of beer and whiskey are determined, inter alia, by enzymes associated with varieties of barley. However, control of varietal homogeneity without expensive bio-chemical tests is still an unresolved problem. One possible approach, using machine vision methods, seems particularly promising due to its potential speed and no need for direct interaction with the grains. The aforementioned difficulty of feature identification becomes even more challenging in the case of barley grains, because they exhibit different features on dorsal and ventral sides.

In this study, we present a machine vision approach to recognition of barley varieties using convolutional neural networks. We propose a neural network architecture with two feature extraction streams, each specialized to process images of a specific side of the grain. This architecture is complemented with a preprocessing recognition step, in order to identify the dorsoventral orientation of each grain. The paper is arranged as follows: in section II we reintroduce a double-stream convolutional neural network from our previous work, in section III we describe the novel architecture, as well as the dataset and training methods, and in section IV we experimentally evaluate performance of the models and compare them.

II. RELATED WORKS

There are several known approaches to barley grain varietal recognition. All of them rely on digital image processing and feature extraction. The features are usually hand-engineered, e.g. edges, texture descriptors, and low dimensional or reduced to a low dimension. Most works also employ some form of machine learning to perform recognition. Zapotoczny *et al.* [1] explore possibilities of classifying images of barley kernels using principal component analysis (PCA), and linear or non-linear discriminant analysis (LDA/NDA). Nowakowski *et al.* [2] use extracted features as learning vectors for an artificial neural network (a multilayer perceptron). Hailu and Meshesha [3] present a classifying ensemble of k nearest neighbors and an artificial neural network. Those approaches yield promising results, but they are only tested on very small datasets (up to several hundred images in up to 5 classes). The scope of work by Szczypiński *et al.* [4] is significantly larger, their dataset comprising over 13,000 images of 11 varieties.

In all of those works, feature extraction and classification are considered separate parts of a system, where the features remain fixed and the classifier is designed using machine learning (ML). Recent advancements in ML made it possible to learn the feature extraction function and classification as a single system. Convolutional neural networks (CNNs) can be trained on raw images, without the difficulty of manually designing the feature extractor. Despite their applications to other agriculture-related problems (e.g. [5]), there have been no attempts to classify barley varieties with CNNs.

In our previous paper [6] we have presented a CNN for detection of defects and impurities in barley grain. Our approach made use of the double-sided imaging capacity of the acquisition system presented by [7]. The double-stream CNN was able to extract features from images of both sides of the grain. Then it fused the feature vectors together, creating a single representation from both images. This enabled it to utilize the information contained within both views of the object to predict its class.

However, due to the unpredictable nature of the imaging process, it was never known which side of each grain was actually visible on which image. Therefore the network had to be robust to this unpredictability, effectively discarding the information about dorsoventral orientation of the grains.

III. EXPERIMENT SETUP

A. Reference neural network architecture

As a reference model we reintroduce the double-stream convolutional neural network from our previous work [6]. This architecture consists of two streams – that is, two separate CNNs – each assigned to a specific camera in order

to process one image of each grain. At some point, depending on the setup details, representations produced by those streams are merged into a single stream. Classification is performed by a feed-forward fully-connected (FC) neural network, whose input is this merged representation.

Dorsal and ventral sides of a grain may exhibit different features. However, during the imaging process the dorsoventral orientation of the grains cannot be constrained. Therefore it is not known which side of the grain is imaged by which camera - the cameras cannot be assigned to any particular viewpoint (i.e. dorsal or ventral). In order to provide robustness against this unpredictability, the two feature extraction networks have shared parameters. That means that even though these are two separate networks with different data flowing through them, their parameters are shared, i.e. constrained to always be equal (fig. 1). Since the camera viewpoints are irrelevant in this setting, we term this architecture viewpoint-ignorant.

The actual CNN implementation used in this study is derived from AlexNet [8], comprising 5 convolutional layers of decreasing kernel size (respectively, 11x11, 5x5, 3x3, 3x3 and 3x3) with 3 overlapping pooling operations between them, and 3 fully-connected layers: first two followed by dropout layers [9], the last one by a softmax operation. After each convolutional and FC layer, a ReLU nonlinearity is applied [8]. The FC layers originally consist of 4096 neurons each except the last one, which is scaled depending on the number of classes. In order to reduce overfitting we limit the capacity of the network by replacing those layers with significantly smaller ones: 64 and 16 neurons each.

A double-stream network is constructed by instantiating two copies of each convolutional layer, although the

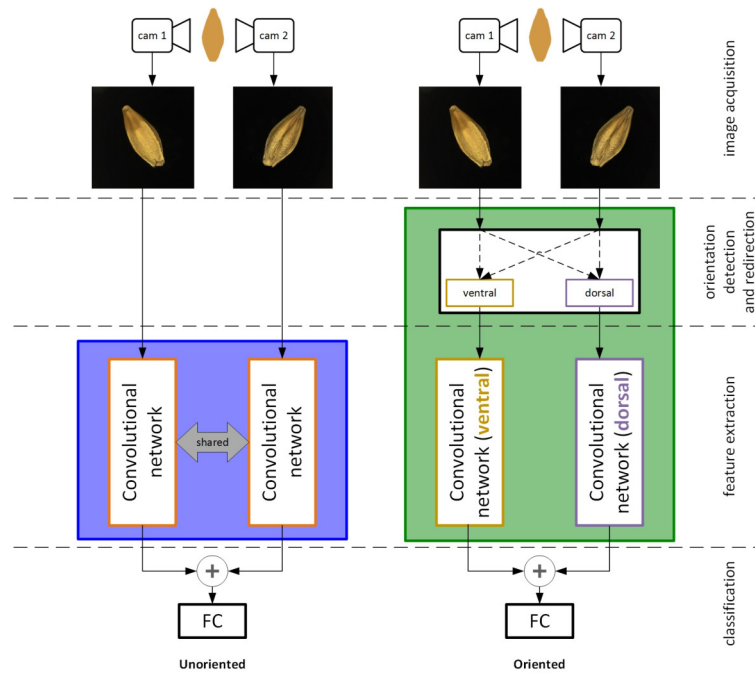


Fig 1. Double-stream CNN architectures: viewpoint-ignorant (on the left) and viewpoint-aware (on the right).

parameters (and gradients thereof) of each pair are constrained to be equal. Each stream will process an image of a different side of the grain, and during training their gradients will be summed. The streams will be merged after the last pooling operation by simple tensor concatenation. The following FC layers will proceed normally, although the input of the first FC layer will be larger to accommodate the concatenated outputs of the streams.

B. Proposed architecture

We propose a network architecture in which the streams are not robust to random dorsoventral orientation of the grains, but instead each learns to extract features specific to each side. In this setting it is assumed that each image the network receives is taken from a particular viewpoint – one of the ventral, another of the dorsal side. We term this network viewpoint-aware.

The novel architecture consists of two streams constructed exactly like in the reference architecture, except for the parameter and gradient equality constraints, which are lifted. This results in two entirely separate feature extraction networks, each associated with a specific view of the object. The rest of the architecture, particularly fusion of the streams by concatenation, and FC layers with softmax, remain structurally identical with the reference architecture.

The assumption that each stream receives a specific view of the object, as opposed to image from a specific camera, requires that the dorsoventral orientation of the object be identified before any processing. In order to achieve this, we introduce a preprocessing step in which the side of a grain is recognized, and the two images are redirected to their associated feature extraction streams according to the result. We solve the task of viewpoint recognition by reducing it to binary classification.

Both the CNN streams and the imaging cameras are ordered, so there is a natural correspondence between them – assume stream 1 is associated with ventral view, then camera 1 can capture either ventral or dorsal side. Therefore, viewpoint recognition becomes a binary classification task: either the images are acquired in the right alignment or not, in which case they need to be switched. For this, we use a CNN of the same structure as the reference architecture with shared streams, except for the final FC layer, which only has 2 outputs (correct alignment or switching needed). The complete system is shown in (fig. 1).

C. Dataset

The data used throughout this research was acquired using a prototype imaging system. The device captured RGB images of individual grains using two cameras located coaxially, opposing to each other, allowing for acquisition of top and bottom views of each grain (details in [7]). Due to the nature of the grain partitioning and transportation subsystem, the dorsoventral orientation of the objects was not predictable. For this reason, no orientation labels could be assigned to the images at the data preparation stage.

Barley was acquired from a research supply. The grains came already separated into 8 varieties, which could be grouped into spring (S) or winter (W), as well as malting (M)

and fodder (F) varieties. There were exactly 2 varieties in each of the group combinations (SM, SF, WM, WF).

A total of 3169 pairs of top/bottom images were acquired, ranging between approximately 200 and 500 pairs per variety. During preprocessing, they were cropped so that the grains were visible in the center of the images, and then resized to 256x256. For the purpose of training and cross-validation, the dataset was split into 3 disjoint subsets. For every cross-validation bin, one of those subsets was used as a training set, while the two remaining ones were merged into a validation set. We applied data augmentation on each training set, appending copies of each image rotated 16 times. Table I shows the dataset composition (pre-augmentation).

TABLE I.
DATASET COMPOSITION

Variety	No. training samples	No. validation samples	Percent of total
SM Bordo	140	278	13.2%
SM Kormoran	163	327	15.5%
SF Mercada	133	266	12.6%
SF Skarb	129	257	12.2%
WM Vanessa	116	232	11.0%
WM Vincenta	166	334	15.8%
WF Kobuz	138	274	13.0%
WF Zenek	72	144	6.8%
Total	1057	2112	100%

D. Training procedure

Neural network training was performed using the Caffe framework [10], with Nvidia DIGITS front-end for task management, using a Nvidia GTX TITAN Z graphics processing unit (GPU) with 2 banks of 6 GB VRAM and 2880 CUDA cores each. To reduce the possibility of overfitting the data, the transfer learning technique was used: each of the convolutional layers was initialized from an AlexNet model pre-trained on ImageNet, a dataset of 1.5 million natural images of various origin in 1000 classes (pre-training, performed independently by Jeff Donahue, BVLC, was not a part of this study). The remaining layers' weights were initialized with Gaussian noise of mean 0 and standard deviation of 0.01, while biases were initialized with a constant of 0.1 each.

Networks were trained using multinomial logistic loss function and Nesterov Accelerated Gradient (NAG) optimization method [11], which is a variation of stochastic gradient descent with momentum. Major training hyperparameters were: momentum $\mu = 0.9$, batch size 128 (as large as could fit in the GPU memory), initial learning rate $\alpha = 0.01$ (as high as the training could still converge at).

For variety recognition, the reference double-stream viewpoint-ignorant network was compared with the proposed viewpoint-aware network. Both networks were trained for 25 epochs: the first 2 epochs at learning rate 0.01, then until epoch 20 at rate 0.001 and for the remaining time at 0.0001. Each process was repeated 3 times for each of the cross-validation folds. Results (F1 measure and confusion

matrices) are reported by averaging of each 3 cross-validation models.

A naïve setup for the viewpoint-aware network would consist of a preprocessing network embedded into the architecture. However, since this sub-network is not being trained at this stage, its presence would only increase the memory and computation power requirement of the entire system, making the training process significantly slower. For this reason, the dorsoventral orientation recognition network was trained separately.

First, a subset of 500 images was selected from the main dataset and annotated manually (with another 500 images selected and annotated for the purpose of validation). The network was trained on this dataset for 20 epochs. After the first 8 epochs learning rate was reduced to 0.001, and after another 8 to 0.0001.

Then, the preprocessing network was queried once over the entire dataset to generate the auxiliary labels containing the information about dorsoventral orientation of each grain. The double-stream viewpoint-aware variety recognition network only read those labels at training time, reducing the preprocessing step only to redirecting images to the feature extraction streams as needed. When using such network in production environment, both the preprocessing sub-network and the variety recognition network would have to be instantiated at the same time.

IV. RESULTS

A. Viewpoint recognition

The dorsoventral orientation recognition network reached 99.8% accuracy after 20 epochs of training. There was little to no overfitting, as both training and validation loss were equal to about -4 in logarithm. The 0.2% error rate was caused by a single image, in which the grain was not imaged from neither the dorsal nor ventral side, but from the sides (fig. 2). This is a rare occurrence which the imaging system allows, but due to an insignificant fraction of such images in the dataset, we decided to ignore their influence – those cases were not handled in any particular way.

B. Variety recognition

Training of a single variety recognition network took approximately 80 minutes (compared to less than 2 minutes for the preprocessing network). All of the models displayed a satisfactory fit – validation loss was actually lower than training, and accuracy was higher (fig. 3). Explanation for this counter-intuitive phenomenon is in dropout. During training, 50% of the fully-connected neurons are randomly deactivated, artificially increasing prediction difficulty, when during validation, no neurons are disabled (their activations are scaled down by a factor of 0.5 to preserve the total magnitude of the activation). This has a significant anti-overfitting effect.

Classification results comparing the viewpoint-ignorant and viewpoint-aware networks, averaged over cross-validation folds, are shown in Table II. Sensitivity and specificity are defined for binary classification, so the table



Fig 3. Difficult case of a sidewise grain orientation (top) versus normal grain exhibiting its ventral (bottom left) and dorsal sides (bottom right)

contains averages of values obtained for each class as a one-versus-all classification.

In fig. 4 we compare confusion matrices for viewpoint-ignorant and viewpoint-aware models. The matrices are normalized row-wise, so a percentage on each tile corresponds to a fraction of images from a given row that were recognized as belonging to the given column (true positive ratio on diagonal). In most cases, the TPR is higher for the viewpoint-aware network – most notably for SF Zenek, an increase from 51.9% to 69.7%. With two classes (SM Bordo, WM Vanessa) the viewpoint-aware network performed worse in terms of TPR. However, in those cases the classification precision (ratio of correct predictions to all predictions as this class, interpreted as probability that a prediction is correct) was significantly higher: 92.80% vs 91.92% for Bordo and 81.43% vs 77.60% for Vanessa.

This confirms that the viewpoint-aware approach is more powerful on average, but the scale of the difference would depend on weights assigned to errors of each kind.

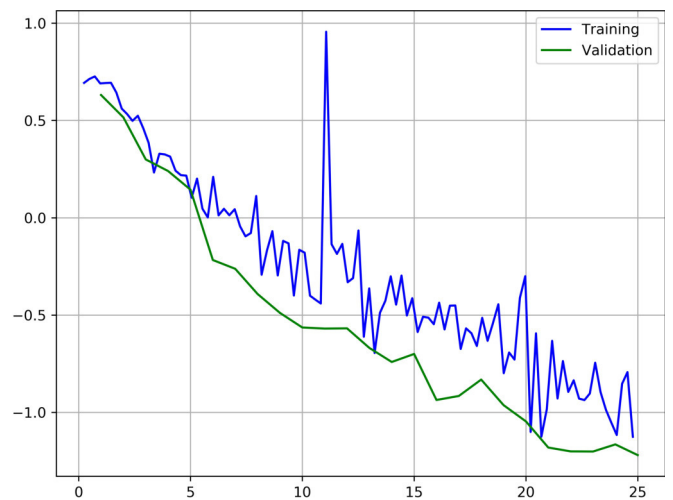


Fig 2. Loss function logarithm on training and validation sets throughout training

TABLE II.
VARIETY CLASSIFICATION RESULTS

Measure	Viewpoint-ignorant	Viewpoint-aware
Accuracy	96.65%	97.24%
Sensitivity	86.63%	88.97%
Specificity	98.09%	98.42%

V. CONCLUSIONS

We have presented a viewpoint-aware double-stream convolutional neural network and proved its superior performance at classification of barley grain varieties. The system performed better on average than a previously shown viewpoint-ignorant network, with slight variations in performance at particular classes.

Those differences might depend on the actual properties of the grains themselves. A detailed study into grain classification would have to account for many such factors, for example phenotypic variability of barley across vegetation seasons.

The system could in principle be trained in an end-to-end setup, if only the dorsal/ventral annotations were available. Due to the image acquisition technique as well as the nature of the imaged objects, obtaining those annotations during data acquisition is not trivial. This is however a limitation of the data imaging system, not our proposed machine learning system.

ACKNOWLEDGMENT

We wish to thank Piotr Lampa and Krzysztof Wall for performing data acquisition.

REFERENCES

- [1] P. Zapotoczny, M. Zielinska, and Z. Nita, "Application of image analysis for the varietal classification of barley," *Journal of Cereal Science*, vol. 48, no. 1, pp. 104–110, Jul. 2008. doi: 10.1016/j.jcs.2007.08.006
- [2] K. Nowakowski, P. Boniecki, R. J. Tomczak, S. Kujaawa, and B. Raba, "Identification of malting barley varieties using computer image analysis and artificial neural networks," presented at the *Fourth International Conference on Digital Image Processing (ICDIP 2012)*, 2012, vol. 8334, p. 833425. doi: 10.1117/12.954155
- [3] B. Hailu and M. Meshesha, "Applying Image Processing for Malt-barley Seed Identification," presented at the *Conference: Ethiopian the 9th ICT Annual Conference 2016 (EICTAC 2016)*, Addis Ababa, 2016.
- [4] P. M. Szczypiński, A. Klepaczek, and P. Zapotoczny, "Identifying barley varieties by computer vision," *Computers and Electronics in Agriculture*, vol. 110, pp. 1–8, Jan. 2015. doi: 10.1016/j.compag.2014.09.016
- [5] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," in *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 452–456. doi: 10.1109/ICIP.2015.7350839
- [6] P. Dolata, M. Mrzyglód, and J. Reiner, "Double-stream Convolutional Neural Networks for Machine Vision Inspection of Natural Products," *Applied Artificial Intelligence*, vol. 31, no. 7–8, pp. 643–659, Sep. 2017. doi: 10.1080/08839514.2018.1428491
- [7] P. Lampa, M. Mrzyglód, and J. Reiner, "Methods of manipulation and image acquisition of natural products on the example of cereal grains," *Control & Cybernetics*, vol. 45, no. 3, 2016.
- [8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [10] Y. Jia et al., "Caffe: Convolutional Architecture for Fast Feature Embedding," 2014, pp. 675–678. doi: 10.1145/2647868.2654889
- [11] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the Importance of Initialization and Momentum in Deep Learning," in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, Atlanta, GA, USA, 2013, pp. III–1139–III–1147.

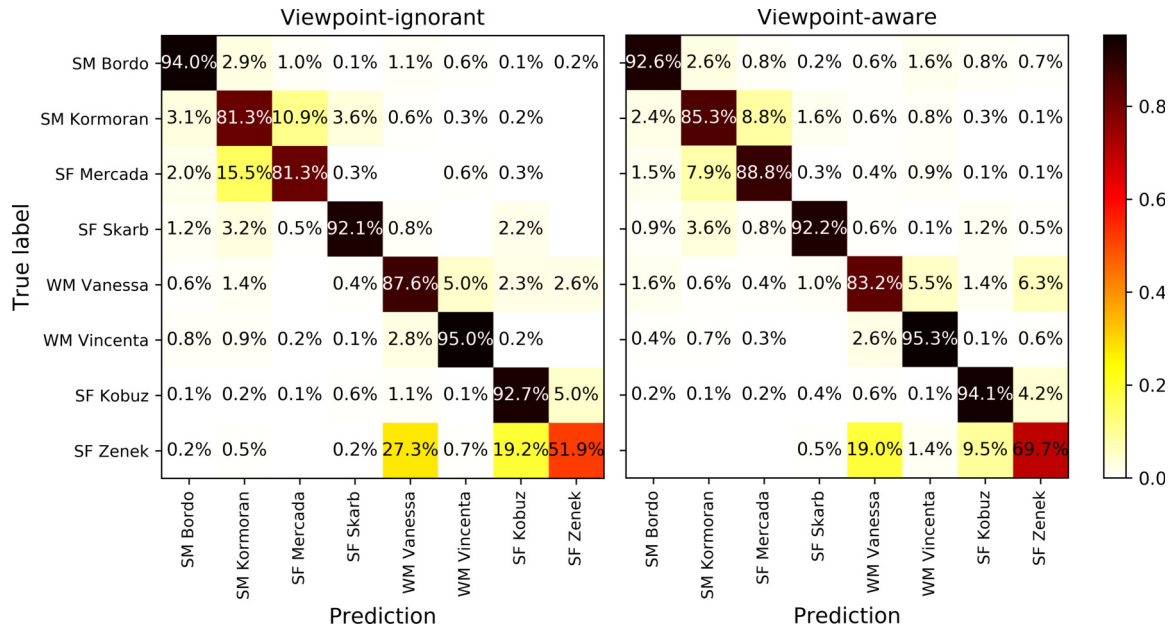


Fig 4. Confusion matrices for viewpoint-ignorant and viewpoint-aware models

Improving Real-Time Performance of U-Nets for Machine Vision in Laser Process Control

Przemysław Dolata

Wrocław University of Science and Technology, ul.
Wyb. Wyspiańskiego 27,
50-370 Wrocław, Poland
Email: przemyslaw.dolata@pwr.edu.pl

Jacek Reiner

Wrocław University of Science and Technology
ul. Wyb. Wyspiańskiego 27,
50-370 Wrocław, Poland
Email: jacek.reiner@pwr.edu.pl

Abstract—Many industrial machine vision problems, particularly real-time control of manufacturing processes such as laser cladding, require robust and fast image processing. The inherent disturbances in images acquired during these processes makes classical segmentation algorithms uncertain. Among many convolutional neural networks introduced recently to solve such difficult problems, *U-Net* balances simplicity with segmentation accuracy. However, it is too computationally intensive for usage in many real-time processing pipelines.

In this work we present a method of identifying the most informative levels of detail in the *U-Net*. By only processing the image at the selected levels, we reduce the total computation time by 80%, while still preserving adequate quality of segmentation.

I. INTRODUCTION

SEGMENTATION of complex, noisy images is a core problem in many industrial applications of machine vision, especially in monitoring and control of laser additive manufacturing processes, such as laser cladding [1]. Where classical image processing algorithms cannot provide necessary robustness (against, for example, plasma emissions or powder scattering), machine-learning-based solutions are applied – recently, convolutional neural networks in particular. However, they are notoriously computationally heavy. For off-line applications this issue can be trivially solved with using more compute power, but in some on-line, real-time applications it is a critical problem. If the process state changes rapidly, any delay in its measurement degrades performance of the control algorithm.

U-Net is a well-known and proven convolutional neural network architecture for image segmentation [2]. Its distinguishing property is a highly modular, symmetric, dual-path structure. In the “down” path, which comprises blocks of max-pooling and convolution layers, features are being extracted from progressively smaller inputs. Those blocks can

be thought of as observing the input at progressively smaller scales. As a result, they produce feature maps with gradually more contextual information, but less spatial resolution. On the other hand, the “up” path integrates the high-context but low-resolution feature maps with intermediate levels of low-context but high-resolution information. This allows producing highly detailed segmentations for objects of different scales.

Training the *U-Net* on laser cladding monitoring images is a relatively straightforward task, even with a small amount of annotated data. The baseline configuration as described by Ronneberger *et al.* [2] outputs segmentations of satisfactory quality without the need to apply any tricks or problem-specific tuning. However, the time of a single image inference, on our in-house hardware, is approximately 250ms. This is unacceptable for any on-line processing purpose – especially for real-time control.

The simplest yet very effective way to decrease processing time is to reduce the size of the input images. This might have an additional benefit of reducing the cost of data acquisition, or allowing higher processing frame rates. A more advanced method would be to downsample the images in the “down” path earlier, skipping some detail scales during inference if the information they contain does not significantly contribute to the overall segmentation quality. However, it is difficult to determine a priori, at which scale should the input be observed and at which intermediate scales should it be processed. Intuitively, this depends on the specific characteristics of a particular problem. Detecting large objects might require more context – hence, deeper “down” path – than small ones. On the other hand, segmenting objects with fuzzy boundaries might not benefit from very high-resolution features as much as when objects have very clear and detailed edges.

In this study, we present a method of determining which blocks in a *U-Net* are really important for correctly segmenting the objects, and which can be removed or skipped to save computation time without significant degradation of prediction quality. Our contribution is primarily a way of optimizing a neural network architecture. However, identifying the levels of detail at which the objects vary can also

This work was supported from the statutory research of Mechanical Faculty of WUST. The source material (images from laser cladding process) for the dataset preparation was supplied by National Centre for Research and Development - Project AmPHora - Additive Manufacturing Processes and Hybrid Operations Research for Innovative Aircraft Technology Development – INNOLOT/I/6/NCBR/2013.

be seen as an important insight, helpful in better understanding of the problem.

II. RELATED WORKS

The original U-Net [2] builds on the concepts of Fully Convolutional Networks [3]. While the FCN allowed using only some of the earlier layers to improve the fidelity of segmentation, U-Net's core concept is to merge even the most early blocks to capture high-resolution features. Further development on these ideas included Pyramid Scene Parsing [4] – where the input is sequentially pooled into separately processed streams and then upsampled and merged together before final prediction – and Feature Pyramid Networks [5], similar to U-Nets except that at every scale a complete segmentation is produced.

Optimization of neural network architectures was always of great interest. Early attempts such as Optimal Brain Surgeon [6] were primarily focused on improving the generalization capability of the learner. In more recent days, most architecture optimization work is focused on improving inference performance or energy efficiency [7], but there are also attempts to use these techniques to help extract classification rules [8]. The two major directions in network structure optimization are: architecture search and network pruning. The objective of architecture search is to find the optimal network structure during training, often using genetic algorithms or growing/pruning strategies [9, 10]. Network pruning focuses on removal of inactive or inefficient units from an already trained network in order to preserve its predictive power but reduce inference time [11].

There is not much research examining the influence of particular levels of detail on the object segmentation or detection quality. Chevalier *et al.* [12] studied the influence of input image resolution on classification performance, however they did not investigate the influence of deeper, highly downsampled layers. In this work, we propose a method of optimizing not only the size of the network input, but also its intermediate levels of detail as well.

III. EXPERIMENT SETUP

A. Scale-specialized blocks

U-Net consists of distinct “blocks”, comprising two 3x3 convolutional layers of various kernel depths, each followed by ReLU nonlinearity. From here onwards we will refer to them as simply *blocks*. Blocks are usually separated by max-pooling (in the “down” path) or upsampling and merge layers (in the “up” path). Thus, different blocks learn to extract features on different levels of detail.

Intuitively, depending on the characteristics of the problem, some of those blocks might be less useful for segmentation. This would mean that features of the data at these levels of detail are not important for a proper recognition. Blocks detecting those features would therefore waste compute power and memory. However, the problem of identifying them is not trivial.

Naively, one could envision training and comparison of multiple networks with different selection of blocks (e.g. one with 3 blocks and downsampling by a factor of 2, or 2 blocks and downsampling by 4). Such a brute-force ap-

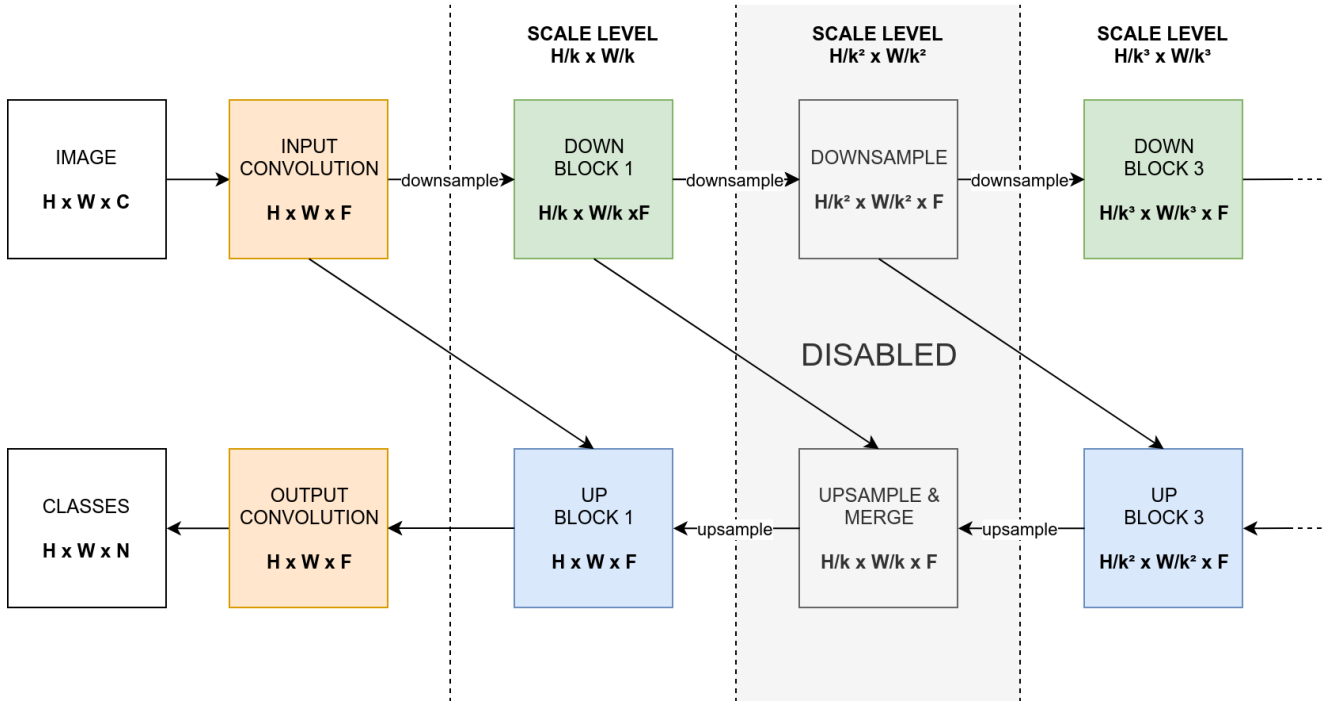


Fig 1. Drop-path regularization algorithm adapted to the general U-Net. In this example, “down” block 1 detects features from data downsampled to some specific resolution (*scale level*), and the corresponding “up” block integrates context extracted the corresponding level. Level 2 is shown in a disabled state – the “down” block only downsamples the data, while the “up” block only upsamples and merges it, both without any other processing.

proach might be infeasible, especially if the dataset is large and the network to be optimized is very deep. Ideally, a single network would be designed and trained in such a way that individual blocks could be freely removed from it without causing a structural failure, but instead only degrading performance – in the case the block was actually useful for prediction. Identifying the useless blocks would then proceed in a manner resembling a structural kind of ablation study.

B. Drop-path regularization

Larsson *et al.* [13] presented a regularization algorithm, *drop-path*, that allowed them to train a very deep, multi-path network so that it behaves like an ensemble of networks. The core idea of drop-path is that if, during every training iteration, a random subset of individual paths in the network is disabled, the rest of the net will be forced to learn to still produce a correct answer. This allows the network to learn robustness against random removal of some sub-paths. Effectively, even though the network trains as a whole, every sub-path tries to become a fully capable standalone predictor itself. Larsson *et al.* report that they were able to extract even a single path of their FractalNet and it still worked almost as good as the whole.

We adapt the drop-path concept to U-Nets in order to allow them to learn robustness against removal of particular *levels of detail*. In a U-Net, the information from a particular scale is utilized twice during a single pass: once in the “down” path, where the features are extracted, and once again in the “up” path where the features are used to improve prediction resolution. Therefore, in our version of drop-path, whenever we randomly disable a “path”, we actually disable both blocks processing data on a particular scale. Overview of the algorithm is shown in Fig. 1.

To allow uninterrupted flow of the data through the disabled blocks, we replace each disabled “down” block with a simple bilinear downsampling layer, and the corresponding “up” block with a similar upsampling layer. We expect the network to learn to segment the images in the absence of information from particular scales, thus allowing evaluation of their influence on segmentation performance by means of a structural ablation test.

C. Simplified U-Net

As the original architecture, U-Net does not naturally accommodate images of every size, requiring cropping and matching between “down” and “up” blocks, depending on the input size. However, as a meta-architecture it is very scalable – one can easily add or remove deeper blocks at different scales in order to capture more or less context in the data. We introduce several changes to the U-Net architecture to simplify it and make it more suitable for the drop-path regularization algorithm.

We add zero padding (1px wide border) to every convolutional layer, making each block preserve its input size. This eliminates the need for complex cropping and matching of data tensors throughout the “up” path.

We set every convolution in every block to produce the exact same number of channels (64), making every layer have exactly the same number of parameters. This is crucial in implementing drop-path: if different blocks produced outputs of different depths (as in the original U-Net), skipping a connection would necessitate a non-trivial mapping between the tensors.

Additionally, we introduce BatchNorm [14] after every convolution layer in order to stabilize the gradients. This is particularly important in the “up” path where data from two separate sources is combined.

Finally, following the practice of FPN, we change the type of connections between the “up” and “down” paths from concatenation (as originally in U-Net) to addition. This forms a residual connection between the paths, similar as described in [15]. This is not a critical change, but it reduces the number of parameters in the “up” convolutions by a factor of two, additionally speeding up the computation.

D. Evaluation by ablation

We expect such a U-Net, trained using drop-path regularization, to behave like an ensemble of smaller networks, each processing data at a particular level of detail. This ensemble should be robust against removal of one member – at most, this should cause the overall performance to degrade, if that member (scale path) strongly contributes to the ensemble’s response. Therefore, we can measure the influence of a particular scale level by a structural ablation study. To test how important a particular level of detail is, we disable its corresponding block and evaluate the network on a validation set, measuring the change in segmentation performance. Additionally, we measure the average inference time to estimate the influence of disabling a block on wall-clock performance of the network.

In the experiment to follow, we use this evaluation strategy to reason about the data – and thus the problem at hand – in two ways.

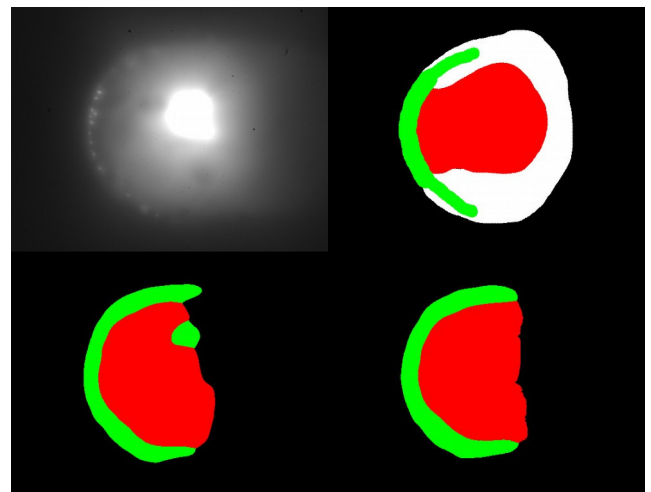


Fig 2. Example data and segmentations. Top row, left to right: source image, ground truth (“ignore” label in white); bottom row: segmentations – left: reduced model (see results, section B), right: full model.

By progressively disabling all blocks starting from the most high-resolution one, we attempt to identify the minimum scale level at which the network can observe the input images while still reliably segmenting the objects. The goal of this experiment is similar to Chevalier *et al.*, except we consider segmentation instead of classification.

By disabling subsequent blocks, starting from some given one, in different combinations, we attempt to find which of the intermediate levels of detail that extract contextual information are actually useful for a correct segmentation. This may provide an insight about how much context and on which level is really necessary, and which levels could be skipped to conserve compute time.

It is important to notice that the initial block of U-Net (on full scale) cannot be disabled – all subsequent layers require the input to be of a certain channel depth, and this first block transforms the original channel to a feature map of a common depth. This means that the initial convolutions will still be performed on the input of original resolution, constituting an approximately constant part of the computation time that cannot be trivially reduced.

IV. RESULTS

A. Reference network

We conduct the experiments on an in-house dataset of images acquired by coaxial on-line monitoring of a laser cladding process. Images obtained during this process are inherently noisy and blurry due to plasma emissions and powder scattering. However, they carry important information about process status, encoded in the shape of the pool of metal molten by the laser beam. The dataset consists of 250 grayscale images 600x600 pixels, manually annotated in 4 classes: background, two object classes of different shape characteristics (“edge” and “pool”) and an ignore label. Data was split in training and validation sets (150 and 100 im-

Parameter	Value
Learning rate schedule	constant 0.01
Adamax momenta	0.99, 0.999
Weight decay	0.0001
Batch size	64
Total iterations	750 000
Drop-path probability	0.25

ages, respectively). Example data and segmentations shown in Fig. 2.

The reference network consists of 5 levels of feature extraction blocks, at following scale levels: 600px, 300px, 150px, 75px, 25px and 5px. Each block comprises two 3x3 convolutional layers with 64 kernels, each followed by a BatchNorm layer and a ReLU nonlinearity. The network was trained using the Adamax [16] optimizer under the cross-entropy loss function. The complete training parameters are given in Table I.

Due to a small number of data samples and the need to train from scratch, heavy data augmentation routine was used in the form of elastic deformations [17] and horizontal and vertical flips. All augmentations were performed on-line in a random manner, directly before feeding data into the network. For testing, the intersection-over-union (IoU) metric was used. Results are given separately for either object class, due to different characteristics of their shapes.

Experiments were conducted in the PyTorch framework [18] using a single Nvidia RTX 2080 Ti GPU for training and an Nvidia TITAN Z for performance testing.

The reference network trained in approximately 11 hours achieving an IoU metric of 0.654 for the “edge” class and 0.809 for “pool” class. The average inference time (with gradient computation disabled) was 154.5ms, which is already approximately 38% faster than the original U-Net.

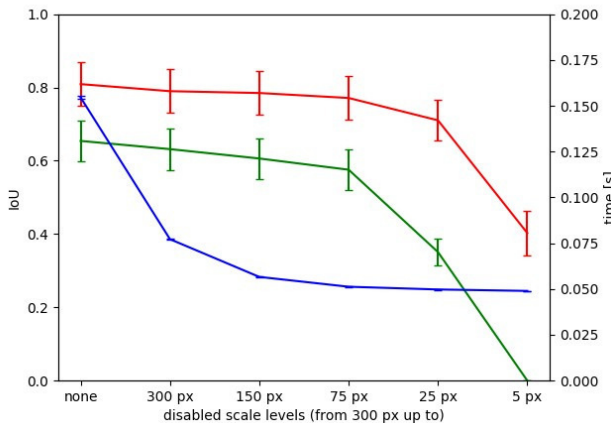


Fig 3. Results of the input size study. Segmentation performance (IoU for both classes, red and green plots) on the left axis, inference time (blue plot) on the right axis.

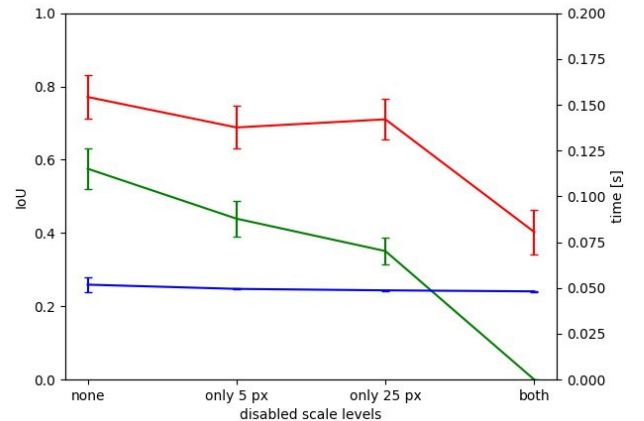


Fig 4. Results of the context levels study. Annotations the same as in Fig. 4.

B. Input size study

Results of the progressive structural ablation study show relatively small degradation of segmentation quality when disabling the early, high-resolution blocks. As illustrated in Fig. 3, removing only the first block cuts the inference time in half, down to 77.2ms while only reducing the IoU score by 0.02. Performance improvements continue to be significant up until the scale levels of 150-75px, saturating at approximately 50ms (80% reduction with respect to the original U-Net). After that point, any potential speed-up is smaller than the cost of repeatedly downsampling the data to the desired resolution, while segmentation accuracy falls rapidly.

Fig. 2 visually compares segmentations produced by a full model (bottom right) and a model reduced by disabling scale levels 300 px and 150 px (bottom left).

C. Context levels study

By disabling blocks at lower scale levels, we can determine the influence of particular context sizes. In this example we disable the first 2 levels (300px and 150px scales), fix the 75px scale as enabled and proceed to disable the remaining scale blocks (25px and 5px) in different combinations. We observe that in our case, disabling any level of context beyond the initial scale of 75px causes a rapid deterioration of segmentation quality, while providing zero practical improvement in inference time. Notice in Fig. 4 how disabling scale level 25px has a much more significant effect on the “edge” class (green) than on “pool” (red). Those most contextual blocks operate at very high relative scales (downsampling by the factors of 3 and 5, respectively) – we surmise that due to this, they learn very independent features that are critical to correct segmentation.

V. CONCLUSION

In this work we presented a simplified and parameterized version of U-Net and adapted the drop-path algorithm to help the network learn as an ensemble of blocks specialized to detect features at specific levels of detail. This allowed us to analyze the importance of individual blocks on the collective network using a structural ablation study. That in turn let us identify blocks that did not contribute significantly to the segmentation, enabling us to make an informed decision to remove them in order to save compute time.

We argue that if the block was deemed unimportant, this might mean that at this particular scale there are no valuable features to be extracted – the data itself contains little valuable information. Therefore, processing inputs at this size is not worth the compute time. Aside from being a useful finding for optimizing a solution, this might also be a valuable insight into the nature of the problem itself.

ACKNOWLEDGMENT

We wish to thank Mariusz Mrzygłód for valuable comments and discussions.

REFERENCES

- [1] W. Rafajłowicz, P. Jurewicz, J. Reiner, and E. Rafajłowicz, “Iterative Learning of Optimal Control for Nonlinear Processes With Applications to Laser Additive Manufacturing,” *IEEE Transactions on Control Systems Technology*, pp. 1–8, 2018. <https://doi.org/10.1109/TCST.2018.2865444>
- [2] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015, pp. 234–241. http://dx.doi.org/10.1007/978-3-319-24574-4_28
- [3] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440. <https://doi.org/10.1109/TPAMI.2016.2572683>
- [4] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid Scene Parsing Network,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6230–6239.
- [5] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944. <https://doi.org/10.1109/CVPR.2017.106>
- [6] B. Hassibi and D. G. Stork, “Second order derivatives for network pruning: Optimal brain surgeon,” in *Advances in neural information processing systems*, 1993, pp. 164–171.
- [7] S. Han, H. Mao, and W. J. Dally, “Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding,” presented at the *International Conference on Learning Representations (ICLR 2016)*, 2017
- [8] H. Lu, R. Setiono, and Huan Liu, “Effective data mining using neural networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 957–961, 1996. <https://doi.org/10.1109/69.553163>
- [9] C. Liu et al., “Progressive Neural Architecture Search,” arXiv:1712.00559 [cs, stat], Dec. 2017. [preprint]
- [10] R. Luo, F. Tian, T. Qin, E. Chen, and T.-Y. Liu, “Neural Architecture Optimization,” in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 7816–7827.
- [11] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, “Pruning Convolutional Neural Networks for Resource Efficient Inference,” presented at the *International Conference on Learning Representations (ICLR 2017)*, 2017.
- [12] M. Chevalier, N. Thome, M. Cord, J. Fournier, G. Henaff, and E. Dusch, “Low resolution convolutional neural network for automatic target recognition,” in *7th International Symposium on Optronics in Defence and Security*, Paris, France, 2016.
- [13] G. Larsson, M. Maire, and G. Shakhnarovich, “FractalNet: Ultra-Deep Neural Networks without Residuals,” arXiv:1605.07648 [cs], May 2016. [preprint]
- [14] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *International Conference on Machine Learning*, 2015, pp. 448–456.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- [16] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.
- [17] P. Y. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” in *Seventh International Conference on Document Analysis and Recognition*, 2003. Proceedings., 2003, pp. 958–963.
- [18] A. Paszke et al., “Automatic differentiation in PyTorch,” in *NIPS-W*, 2017.



Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Instance segmentation of root crops and simulation-based learning to estimate their physical dimensions for on-line machine vision yield monitoring

Przemysław Dolata^{a,*}, Paweł Wróblewski^a, Mariusz Mrzygłód^a, Jacek Reiner^a^a Wrocław University of Science and Technology, Poland

ARTICLE INFO

Keywords:

Yield estimation
Machine vision
Machine learning
Image segmentation
Synthetic dataset

ABSTRACT

Modern agriculture is based on control and optimization, where monitoring is essential. But yield monitoring limited to spatial mapping of biomass is unsatisfactory for uniformity optimization. Acquiring this information as soon as during the harvest could improve efficiency of precision agriculture. Current machine vision solutions do not allow this for several reasons, starting with difficulties with recognition of heavily cluttered, often mutually occluding objects, or requirements for elaborate hardware. Acquiring images in on-line conditions often results in perspective distortion which makes measurement difficult. This is even more challenging in the case of root crops, as they are usually graded basing on their minimal diameter, which is not directly observable via monocular vision.

We propose a method for developing yield monitoring systems that can estimate physical dimensions of crops directly on-line. In our approach, individual plants are segmented using Mask R-CNN, allowing clear separation even in heavily cluttered conditions and in the presence of occlusions. Then, a nonlinear regression model is used to predict the minimal diameter of each plant basing on their observed contours. This model is capable of jointly correcting the perspective distortion and estimate the non-observed dimension. Training this model requires per-object annotations with their true diameters. Since this information is very difficult to acquire for real crops, we build a simulation environment where we model the crops, eventually rendering a synthetic dataset to train the regression model on. The size estimation model transfers the knowledge from simulated data, correctly predicting the sizes of real plants. We demonstrate our method on the case of potatoes, but it can be applied to other crops as well.

1. Introduction

1.1. Background

Yield monitoring is an important area of agricultural research and practice. Information on crop growth, quality, and yield constitutes vital feedback for planning fertilization and seeding strategies. In other uses, the yield information can be a guiding factor in research. Machine vision-based methods of yield monitoring are receiving increasing attention, because of their potential for easy integration with existing harvesting machines – without the need of significant hardware modifications.

In some applications, simply counting the produce or estimating its total mass is not enough. Root crops, for example, are commercially graded basing on the physical dimensions of individual tubers, among other factors. Plants with different intended purpose must meet different sets of standard criteria. Ensuring uniformity of produce quality within a batch is important. Therefore, acquiring the size information sooner, preferably during the harvest, could improve the efficiency of the harvest.

Harvester-based imaging solutions must overcome several technical difficulties that are not present in a laboratory setting. Hardware conditions limit the possibilities for installation of imaging sensors and additional illumination. Environmental conditions such as sunshine,

* Corresponding author at: Department of Laser Technology, Automation, and Production Organisation, Faculty of Mechanical Engineering, ul. Łukasiewicza 5, 50-370 Wrocław, Poland.

E-mail addresses: przemyslaw.dolata@pwr.edu.pl (P. Dolata), pawel.wroblewski@pwr.edu.pl (P. Wróblewski), mariusz.mrzyglod@pwr.edu.pl (M. Mrzygłód), jacek.reiner@pwr.edu.pl (J. Reiner).

<https://doi.org/10.1016/j.compag.2021.106451>

Received 28 July 2020; Received in revised form 13 July 2021; Accepted 7 September 2021

Available online 27 September 2021

0168-1699/© 2021 Elsevier B.V. All rights reserved.

dust or debris make sensing with complex devices, such as time-of-flight cameras or laser scanners, either difficult or infeasible. On the application side, relying on such expensive laboratory equipment will likely inhibit adoption of such systems in agricultural practice.

Some information about the crops can be, however, extracted from images acquired using simple RGB cameras. Mounting such a simple system directly on the harvester is simple, but the real challenge concerns analyzing the acquired images. Due to the nature of the harvesting process and the way crops are transported on the harvester transmission belt, they often resemble a stream of dense mass rather than a series of separate objects. While it is relatively straightforward to distinguish between the mass of crops and the background, lack of clear boundaries between the crops makes it difficult to recognize individual objects. Performing two-dimensional per-instance measurement necessitates not only recognizing them (predicting bounding boxes around the objects of interest – *object detection*) but also extracting detailed contours of the touching objects (*instance segmentation*).

Moreover, imaging a heavily cluttered scene using a single sensor means that a complete three-dimensional representation of the detected objects cannot be reconstructed. Consequently, if a measurement of some particular dimension is desired – as is the case with potatoes, for instance, where the minimal diameter is of interest – the required information is not directly available.

Another difficulty is related to perspective, which distorts the perception of the scene. When imaging three-dimensional objects, the typical simple approach to perspective correction – based on camera calibration and affine transformation – fails. In these cases, the often-followed practice is to install the imaging system directly overhead the scene. However, such a setting might not allow measurement of the minimal diameter of oblate objects that align themselves flat on the imaging plane.

Viewing the scene at an inclined angle can increase the probability of capturing of the minimal diameter but increases the effect of perspective distortion and occlusion. It means that when the objects appear on the scene in a dense stream over a flat surface, imaging at an angle will cause the objects located closer to the sensor to occlude those located further. This is an important factor, because measuring only the directly visible parts of the interesting objects will necessarily result in underestimating their sizes.

1.2. Related work

Reasoning about crops using machine vision is often challenging due to the inherent variability of plant products, owing to their biological nature. There exists a large body of knowledge in this area, with varying degree of complexity of the specific problem approached. The general trend is that more information can be extracted from images acquired in very good conditions, while field-based imaging systems usually only allow a more basic type of analysis, such as counting of fruits. Most of the works on the subject of precise crop characterization are done in idealized conditions, often using specialized hardware. For example (Khojastehnazhand et al., 2019) describe a system for volume estimation and ripeness classification of apricots, in which single fruits are placed on a highly contrasting background and imaged using a custom system comprising a CCD camera and LED illumination installed in an enclosed chamber. Similar machine vision measurement systems are shown in (Momin et al., 2017) for mango fruits and in (Zhang et al., 2018) for betel nuts.

Quality inspection and grading solutions for agriculture basing on machine vision were known at least since (Noordam et al., 2000) who used line-scan cameras and mirrors to obtain images of whole potatoes at high speed. Many current studies are also based on custom imaging systems, often utilizing more advanced sensors to retrieve information about the objects. (Kusumam et al., 2017) built a tractor attachment housing an RGB-D camera and LED illuminators, allowing not only to identify and locate broccoli heads in the field but also to estimate their

size. In a more controlled environment, (Wang et al., 2016) demonstrated the use of laser triangulation to reconstruct the surface of diced potatoes in order to measure and grade them. Other examples include (ElMasry et al., 2012; Su et al., 2018; Long et al., 2018), but commercial solutions are also available for post-harvest grading and sorting, similarly based on purpose-built imaging systems for highly detailed analysis – e.g. *Ellips TrueSort*.

The approaches basing on specialized imaging solutions cannot however be easily deployed in field conditions due to the risk of damage or the presence of dust, adversely influencing the highly sensitive equipment. That is why some research has been devoted to applying simple and inexpensive hardware to solve grading, measurement or quality inspection problems. However, when only simple cameras with no additional illumination are available, the identification of objects of interest becomes difficult. Thus, some works assume very good imaging conditions. For example, (Pandey et al., 2019) present an approach to grading and defect detection in potatoes but assume that their source material is easily distinguishable against the contrasting background and there are no significant occlusions between the objects. (Hofstee and Molema, 2003) managed to estimate volume of potatoes, dealing with separating clusters of objects in greater detail, but again omitting the problem of segmentation or occlusions. (Razmjoooy et al., 2012) also perform geometric measurements in the case when the objects are imaged in separation and highly contrast with the background.

Field conditions incur limitations not only on the hardware one can deploy, but also when each plant species can be imaged. Pre-harvest observations of fruit trees in order to estimate crop load are common, and the typical problem encountered in this area is segmentation of the fruits or other interesting parts of the plant against the stems and leaves (Lin et al., 2020). In another study, (Lee et al., 2018) distinguish potatoes from the underlying soil in order to estimate their mass. However, this problem is rarely encountered in practice, as potatoes are not naturally observed in such conditions. Other works related to detecting and counting crops in the field include also e.g. (Chen et al., 2017).

Many of above works start with a certain other assumption that simplifies image processing: that the objects rarely touch or overlap one another. With some crops however this is not the case: for example, many root crops. Because they are not visible before the harvest, the first opportunity to image them is during that process. However, when moving through the harvester pipeline, the objects tend to clutter, and distinguishing between individual instances becomes a challenge. This is shown in (Boatswain Jacques et al., 2018) who installed an RGB camera on an onion harvester to count shallot onions. While their solution for segmentation of onions against the harvester hardware works fairly well, they struggled with identifying the heavily crowded objects. Similar difficulties were encountered by (Chinchuluun et al., 2009) with counting of citrus fruits, despite integrating a custom housing containing a camera and illumination with the harvester.

Recent years and the success of convolutional neural networks (CNNs) brought solutions based on deep learning to precision agriculture. These methods rely on the concept of feature learning, as opposed to extracting a series of predefined features, e.g. using MaZda (Szczyński et al., 2009). Approaches based on deep learning are often applied in the area of fruit detection and counting. (Bargoti and Underwood, 2017) developed an apple counting system based on a CNN with a custom solution for separating the touching objects. It should be noted however, that the apples in their images are not as heavily cluttered as e.g. onions on a harvester. Similar approach with a similar assumption about the object separation can be seen in (Kestur et al., 2019). (Habaragamuwa et al., 2018) approach a much more difficult problem of detecting strawberries that often occlude one another. They employ a region-based convolutional neural network (R-CNN) that returns a bounding box for each detected object, allowing to separate even partially occluded fruits. (Yu et al., 2019) take strawberry detection a step further and segment individual objects using Mask R-CNN. (Sa et al., 2016) propose a detection framework for fruits based on Faster R-

CNN, called DeepFruit, that is simple to apply for other species.

Those approaches are almost universally applicable only to the problems of detection, which allows counting of produce, but provide limited information on their size. When a more precise measurement is desired, CNN-based approaches are not always employed. For example, (Si et al., 2018) developed their own algorithm to acquire information on the shape of potatoes moving on a conveyor belt at a high speed, accounting for a considerable motion blur resulting from using a simple imaging system with no additional illumination. When CNNs are used to detect objects in order to perform geometric measurements, as for example done by (Yu et al., 2020) in the case of fish, the imaging conditions are selected to facilitate simple measurement. In this and above works, the small scale of the imaging scene and the right angle between the optical axis and the imaging plane ensure that perspective only negligibly influences the measurements. Calibration in such cases can be conducted simply by computing a constant scale factor between the image space and the physical space.

For a broader context on machine vision in agriculture, one can consult the surveys (Gongal et al., 2015; Mavridou et al., 2019).

Among the studies done on yield monitoring using machine vision, all works that we are aware of base on at least one of the simplifying assumptions: either employ specialized and expensive hardware to acquire images, or perform the imaging in idealized conditions, or the imaged plants are mostly well-separated, or the objects are not measured in detail but only detected. To the best of our knowledge, there are no approaches to yield monitoring performed in the field conditions (on a harvester) with minimally invasive hardware, able to provide size information on the highly cluttered crops.

1.3. Aims

We propose a method for counting and estimating the size of root crops during harvest basing on images acquired using a simple RGB camera mounted directly on the harvester, without any additional hardware. The approach consists of two steps: object detection and size estimation. We perform the object detection and instance segmentation using a convolutional neural network, specifically a Mask R-CNN. This framework allows good detection rate even in difficult conditions and can also detect and segment objects that are occluded to some extent. We propose a nonlinear regression model for estimation of the physical size of the object. Since calibration of this model requires reference data, which is not available for the real-world problem, we construct a simulation environment in order to generate it. Creating a three-dimensional model of the crop allows us to render a synthetic dataset for calibration of the regression model. The proposed method is independent of the particular crop, but its particular implementation is crop-specific; therefore, we demonstrate the results for the case of potato (*Solanum tuberosum*).

2. Materials and methods

2.1. Outline of the method

Our proposed method consists of several steps. In order to estimate physical dimensions of objects from images, first these objects need to be detected and individually segmented. We employ instance segmentation techniques basing on convolutional neural networks (CNN) for this purpose. In the next step, the physical dimensions of the objects need to be estimated from their image-space contours. To solve this, we implement a neural-network-based regression model. However, this model requires a constant-length representation of the input. Contours obtained in the instance segmentation step are variable-length, therefore we approximate them using ellipse fitting. Fig. 1 (blue pathway) summarises the concept.

Both crucial elements of the system are basing on machine learning, therefore annotated data is needed to train them. We train the instance segmentation model using manually annotated images of real potatoes, acquired in field conditions. However, the same is not possible for the physical dimension regressor, as it is infeasible to directly measure every single potato tuber. To overcome this challenge, we propose a simulation environment for rendering “synthetic” images of potatoes, equipped both with segmentation masks as well as exact physical dimensions of the displayed objects. The potatoes are modelled to closely match the real tubers, according to some statistical distribution describing their shape. This way the image-space contours can be matched with the corresponding physical-space dimensions, enabling training of the regression model (green pathway on Fig. 1).

2.2. Image acquisition

We acquire images of potatoes on a harvester in the field conditions using a simple 4 megapixels RGB camera (RunCam 2, RunCam Technology Co.) housed in a custom 3D-printed case with integrated microcontroller and GPS receiver. The imaging system was fastened to the frame of the harvester, as shown in Fig. 2. To calibrate the images, we applied a standard checkerboard-pattern calibration routine. The complete system was configured to acquire images and geolocation meta-data at a rate of approximately 1 image every 5 seconds, storing them on an SD card.

Images undergo no specific preprocessing before being fed to the instance segmentation CNN. The only additional work beyond acquisition is manual annotation of the images. A custom in-house annotation software is used to draw contours around every object, including the outlines of their occluded fragments.

2.3. Instance segmentation

Our approach to instance segmentation is based on a convolutional neural network (CNN), following the Mask R-CNN framework (He et al., 2017). Although not dedicated for dealing with heavy occlusions, we

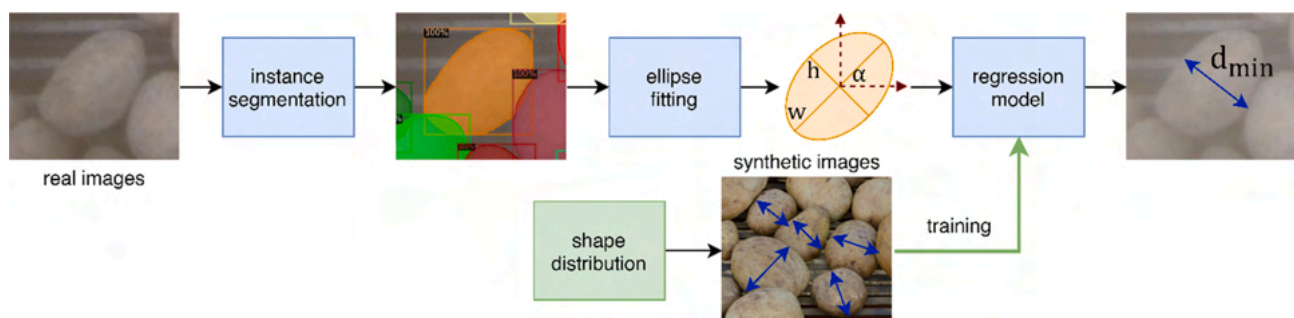


Fig. 1. Outline of the proposed approach.



Fig. 2. Field experiment. Image acquisition system deployed during harvest (left); close-up on the system enclosed in a pink 3D-printed case (right).

found it to be sufficiently capable of segmenting moderately overlapping objects. For details of the architecture, we refer the reader to the original paper. We make no structural changes to the default architecture, using the implementation provided in Detectron 2 toolbox (Wu et al., 2019). This toolbox is built on top of PyTorch framework (Paszke et al., 2017), which we also base our research code on. The specific architecture used throughout this paper follows the built-in model mask_rcnn_R_50_FPN_3x – using ResNet-50 (He et al., 2016) as the feature extraction backbone. To initialize our models, we use pre-trained weights from the Detectron 2 model repository (“model zoo”): ResNet-50 feature extraction backbone has been pre-trained on the ImageNet-1k, while the instance segmentation model on the MS COCO dataset.

We train our instance segmentation models using the standard Stochastic Gradient Descent (SGD) optimization algorithm. To make reproduction of our results easier, we leave all of the hyperparameters at their default values except where noted. In order to artificially increase the dataset size and improve robustness of the model, we apply basic data augmentations during training. In order to preserve all of the contextual information in our images, we refrain from using highly distorting augmentations such as elastic deformations. Instead, we only apply random horizontal and vertical flips and random scaling (from the input crop of 800 pixels to a random size between 700 and 1000 pixels).

Quality evaluation of instance segmentation algorithms is often performed using the Panoptic Quality measure (PQ) (Kirillov et al., 2019). PQ measure multiplicatively combines two separate terms: Recognition Quality (RQ), which is an F1-score of correctly/incorrectly predicted bounding boxes, and Segmentation Quality (SQ), expressed in terms of the average intersection-over-union (IoU) over all correctly detected objects. The metric is given by the following formula:

$$PQ = \frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \cdot \frac{\sum_{\text{sample} \in TP} \text{IoU}(\text{sample})}{|TP|} \quad (1)$$

in which the first element constitutes RQ and the second – SQ. Whenever we report the overall PQ metric, we also give the values of its components, as they carry different information. Low RQ means that the detection system fails to recognize all of the objects present in the image or outputs false positives, which are penalized equally with false negatives. Low SQ informs that the *shapes* of the correctly recognized objects are not properly predicted. The latter will influence the size estimation quality.

Since the PQ measure as implemented in Detectron 2 assumes that all detected objects are separate, we implement a custom version of the

metric to allow measuring overlapping objects as well.

2.4. Learning to estimate the physical dimensions

2.4.1. Geometry of the imaging system

Instance segmentation provides information on each visible (detected) object as a contour in the image space. The task of estimating the true dimensions of the corresponding object in the physical space is nontrivial for two reasons. First, the images of objects are significantly distorted by the perspective projection. Simple correction via perspective transformation allows performing accurate geometrical measurements only when the measured objects lie on the imaged plane. Objects that protrude from the imaged plane will be unnaturally distorted by this transformation, making their measurements incorrect. This idea, in a one-dimensional simplification, is illustrated in Fig. 3. Second problem lies in the fact that in order to grade root crops, the minimum diameter $d_{\min}$ needs to be obtained. Due to the behavior of the physical objects on the transporter, this diameter is not oriented with respect to other dimensions in any predictable way, making its direct measurement impossible. The information obtained by imaging is the diameter in the image space, d_{px} , expressed in pixels. Using a simple perspective transformation to the imaged plane would yield the projected size d_{proj} (in

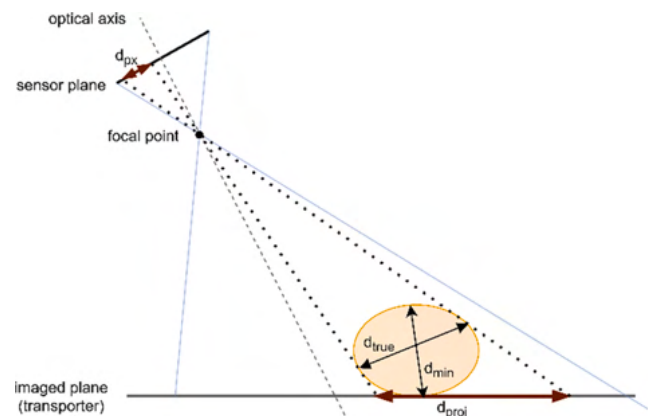


Fig. 3. Geometry of the imaging system. The back-projected dimension d_{proj} is not the same as the actual object diameter, d_{true} . d_{true} , in turn, is not necessarily equal to the minimal object diameter, $d_{\min}$.

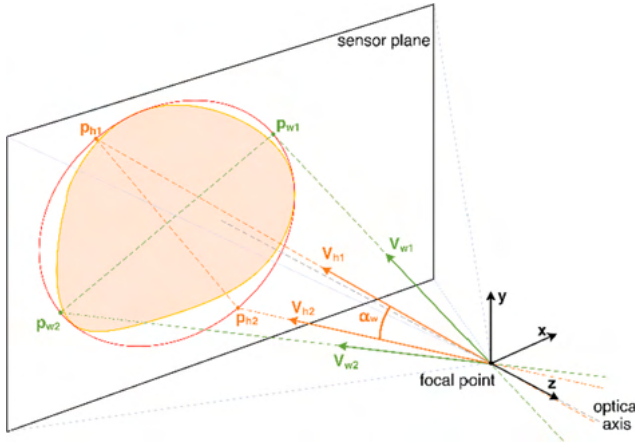


Fig. 4. Extreme points of the ellipse in the camera space.

world dimensions, e.g. millimeters). This is however not the diameter that has generated the imaged projection, d_{true} . The relationship between d_{true} and d_{px} depends not only on the geometry of the imaging system, but also on the shape of the three-dimensional object itself. The image alone does not contain information sufficient to exactly calculate neither d_{true} nor the minimum diameter d_{min} .

2.4.2. Ellipse fitting and angle-based description

As previously mentioned, we parameterize the detected contours by ellipse fitting. We use the `fitEllipse` implementation from the OpenCV toolbox, which returns the ellipse parameters in terms of position of its center, length of its axes, and its rotation angle β . However, estimating physical dimensions from a pixel-based representation in the image space suffers from a distortion problem.

We circumvent this by transforming the ellipse parameters into an angle-based representation in the frame of reference of the camera (*camera space*). Following a standard camera calibration procedure, we obtain the matrix K , which enables transformation of a point p_{img} in the two-dimensional image space into a point p_{cam} in the three-dimensional camera space:

$$p_{cam} = p_{img}K \quad (2)$$

This in turn allows representing the ellipse in terms of its angular height and width via a simple procedure (illustrated in Fig. 4): compute extreme points p_{w1} , p_{w2} , p_{h1} , p_{h2} of the ellipse in the image space, transform these points into vectors $\bar{v}_{w1}$, $\bar{v}_{w2}$, $\bar{v}_{h1}$, $\bar{v}_{h2}$ in the camera space, and finally compute angles between the resulting vectors. Formulae are included in Table 1.

This angular representation is not enough to correctly estimate the size of the objects, however. Information on the relation between the camera space and the real world space is crucial. We provide it in the form of the angle θ defined as the angle between vector $\bar{v}$ representing the center of the object in the world space and vector $\bar{n}$ normal to the plane of the transporter (Fig. 5). The reason for selecting this angle is twofold: first, embedded within this angle is the approximate distance

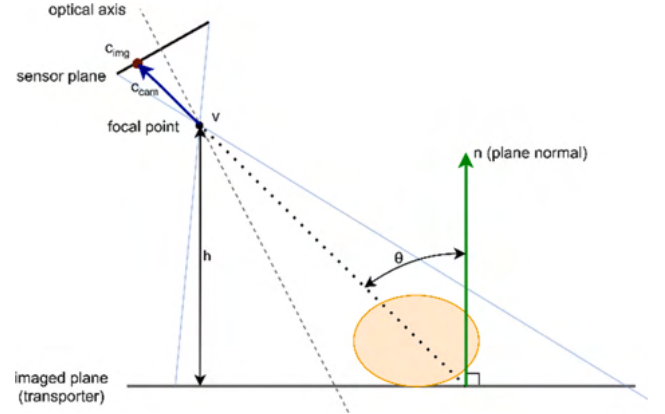


Fig. 5. Calculation of θ in the camera space.

Table 2

Structure of the diameter regression model.

Layer	Inputs	Outputs	Number of parameters
Linear	4	4	20
ELU ($\alpha = 1$)	4	4	n/a
Linear	4	3	15
ELU ($\alpha = 1$)	3	3	n/a
Linear	3	1	4

between the object and the camera, providing the necessary information on scale; second, this angle provides information on direction of observation of the object, which we assume to influence the relation between its observed and real dimensions.

To compute θ , we first conduct the standard calibration procedure involving a checkerboard pattern placed on the surface of the transporter. This procedure allows finding $\bar{n}$ in the camera space. All calculations can thus be performed on camera-space vectors ($\bar{c}_{cam}$, $\bar{n}$).

In order to fully separate the size estimation from the geometry of a specific imaging system, we introduce a scaling factor h . All of the computations are performed in a coordinate system normalized so that $h = 1$, and later multiplied by the original factor to return to the physical coordinates. This allows deployment of the estimation system in any acquisition system of a known geometry without the need to retrain the regression model.

2.4.3. Regression model for diameter estimation

We have established that the information required to exactly measure d_{min} is not contained within the image. However, we propose that it is possible to statistically estimate this diameter if assumptions can be made regarding the shape of the object. We also propose that a nonlinear model can predict real dimensions of an object in the presence of distortions that a standard linear transform could not rectify. We propose a single regression function f to estimate the minimal diameter d_{min} :

$$d_{min} = h \cdot f(\bar{x}, \varphi) \quad (2)$$

Table 1

Features constituting the input vector for the diameter estimation model.

Symbol	Description	Formula
α_h	Angular height of the ellipse (tangent of the angle)	$\alpha_h = \tan\left(\frac{\cos \bar{v}_{h1} \cdot \bar{v}_{h2}}{ \bar{v}_{h1} \bar{v}_{h2} }\right)$
α_w	Angular width of the ellipse (tangent of the angle)	$\alpha_w = \tan\left(\frac{\cos \bar{v}_{w1} \cdot \bar{v}_{w2}}{ \bar{v}_{w1} \bar{v}_{w2} }\right)$
β	Rotation angle of the ellipse (sine of the angle)	Direct output of <code>fitEllipse</code>
θ	Viewing angle of the object (tangent of the angle)	$\theta = \tan\left(\frac{\bar{v} \cdot \bar{n}}{ \bar{v} \bar{n} }\right)$

where $\bar{x}$ is a vector of input measurements (described in detail in Table 1), and model parameters vector ϕ implicitly describes the perspective distortion and the shape distribution of the objects.

Our nonlinear regression function f is constructed as a feed-forward fully connected neural network. The structure consists of three dense (following PyTorch terminology: *linear*) layers interleaved by two ELU activation functions. Details of the structure are explained in Table 2.

We train this model using gradient descent, optimizing a mean squared error (L2) loss function with the Adam optimizer with constant scaling coefficients of 0.9 and 0.999 and an L2 weight decay. Due to the small size of the model, we do not use mini-batches, instead optimizing on the entire dataset each iteration. Table 3 summarizes the training parameters and their scheduling.

Dataset used in training consists of a series of feature vectors – obtained by detecting objects using Mask R-CNN and fitting ellipses to their contours – with corresponding true values of each object's minimal diameter. The model is trained on one subset of the data and validated on the other subset using the R^2 measure to quantify the goodness of fit.

2.5. Simulation-based learning

2.5.1. Synthetic dataset generation

Images acquired using a simple optical system installed on the harvester do not provide any information on the true sizes of the objects, due to the technical infeasibility of performing accurate measurements on a per-object basis directly during imaging. Therefore, no data exists to train the size estimation model. To circumvent this problem we create a simulated environment in order to render synthetic images containing any distribution of objects, at the same time providing object-level labels not only in the image space (contours for segmentation) but also in the physical domain (true dimensions). This environment needs to be modelled in a way that closely resembles the underlying physical phenomena. In this subsection, we distinguish four key aspects of this resemblance and briefly analyze their importance: the object of interest, physics describing behavior of the objects in the environment, camera and optics, and rendering quality.

The object of interest – in our case potato tubers – is the single most important component. Its shape needs to follow the distribution of the real potatoes. We follow (Torppa et al., 2007) and model our objects according to the shape coefficients (mean elongation, mean flatness, and the corresponding standard deviations – for details, see Table 1 in the original paper) for the Van Gogh cultivar, version 1. For the irregularities in the shape of potatoes however, we do not implement the proposed spherical-harmonics component, instead creating a custom random deformation model basing on a series of reference meshes. Those are then deformed to follow the size distribution mentioned above. The minimal diameter – which can be thought of as a scaling factor for the synthetic potatoes – follows a normal distribution with mean 55.0mm and standard deviation 10.6mm. These values allow the distribution to cover all possible sizes of potatoes, while including less than 3% of tubers so small that they would be rejected by the internal sieve on the harvester (diameter less than 35mm).

Physics, in particular collisions between potatoes and their rolling behavior on the transporter, needs to be realistically modelled. This is important to correctly simulate the occlusion conditions that occur in the real-world harvest scenario. Our simulation environment is built using Blender (Blender Online Community, 2019), owing to its

capability of simulating highly detailed physics as well as integrated support for Python programming language.

Due to the angle-based representation of the objects at the size estimation stage, the camera settings are unimportant, as long as sufficient information can be extracted to allow calculation of the viewing angle for each pixel in the image. However, for reasons that will be discussed separately, we set the virtual camera to ensure that the output resolution as well as the relative scale of rendered objects match those of the real dataset.

Finally, the rendering quality is also an important factor, as differences between the visual style of the resulting synthetic images and the real acquired images might cause errors on the object detection stage. Many computer vision algorithms – including CNN-based – do not work equally well across different *domains*, meaning that a detector trained on real data might not predict well enough on synthetic data, and vice versa. There is a growing body of research done in the directions of *domain adaptation* and training data synthesis (Barth et al., 2018), but experimentally confirming these approaches in the context of instance segmentation of root crops is beyond the scope of this work. Alternatively, one could build a photorealistic rendering pipeline, but we deem this approach infeasible in practice due to the complexity involved. Instead, we propose a pipeline based on simple rendering.

2.5.2. Simple rendering and dual segmentation pipelines

Rendering images is not the main reason behind building a simulation environment. The actual objective is to generate a dataset suitable for training the size estimation model, consisting of image-space contours equipped with physical-space dimensions. The generating physical process needs to closely match that of the real potatoes, so that the resulting contours are indistinguishable from those produced by an instance segmentation algorithm working on real images. However, the actual rendered images do not need to appear visually similar to the real ones. In a way, images can be thought of as just a representation intermediate between a physical object and a detected contour.

Therefore, we render our images using simple settings, but we employ a separate instance segmentation model to detect objects on these images. This “synthetic” segmentation model is incompatible with the “real” model in terms of image-space inputs. But as long as errors in detection and segmentation between the two models are similar, we can use the contours produced by the “synthetic” model in order to train the size estimation regressor, and expect this regressor to produce correct predictions even when it is fed contours produced by the “real” model.

In order to assess the error similarity between the two models, we consider their PQ measures (specifically, recognition and segmentation components separately) on their respective validation sets. The desired result is that both models achieve similar RQ and SQ measures, meaning the models predict similar contours given similar inputs (we dub this “PQ-correspondence”, after the measure). Ensuring that this is the case is critical for the development. However, it might so happen that one of the models will outperform the other. We recommend applying a corrective strategy such as early stopping of the training process to prevent this from happening. More elaborate strategies could be object of further research.

The detailed map of our approach is shown in Fig. 6. First, the instance segmentation model for the real images is trained and validated (1). Then, the simulation environment is built, with 3D models of the objects matching the shapes of the real objects (2). Next, the instance segmentation model for the rendered images is trained and validated (3) in such a way that prediction errors are similar to those produced by the real-data model (4). Finally, the regression model for size estimation is trained on data produced by the “synthetic” pathway (5). Consequently, basing on correspondence in the physical domain (accurate modelling of potato tubers) and PQ-correspondence (prediction errors of the instance segmentation models), this size estimator can be transferred to the “real” pathway.

Table 3
Training hyperparameters schedule for the regression model.

Stage	Epochs	Learning rate	Weight decay
1	250	0.01	0.001
2	500	0.001	0.0001
3	750	0.00025	0.00001

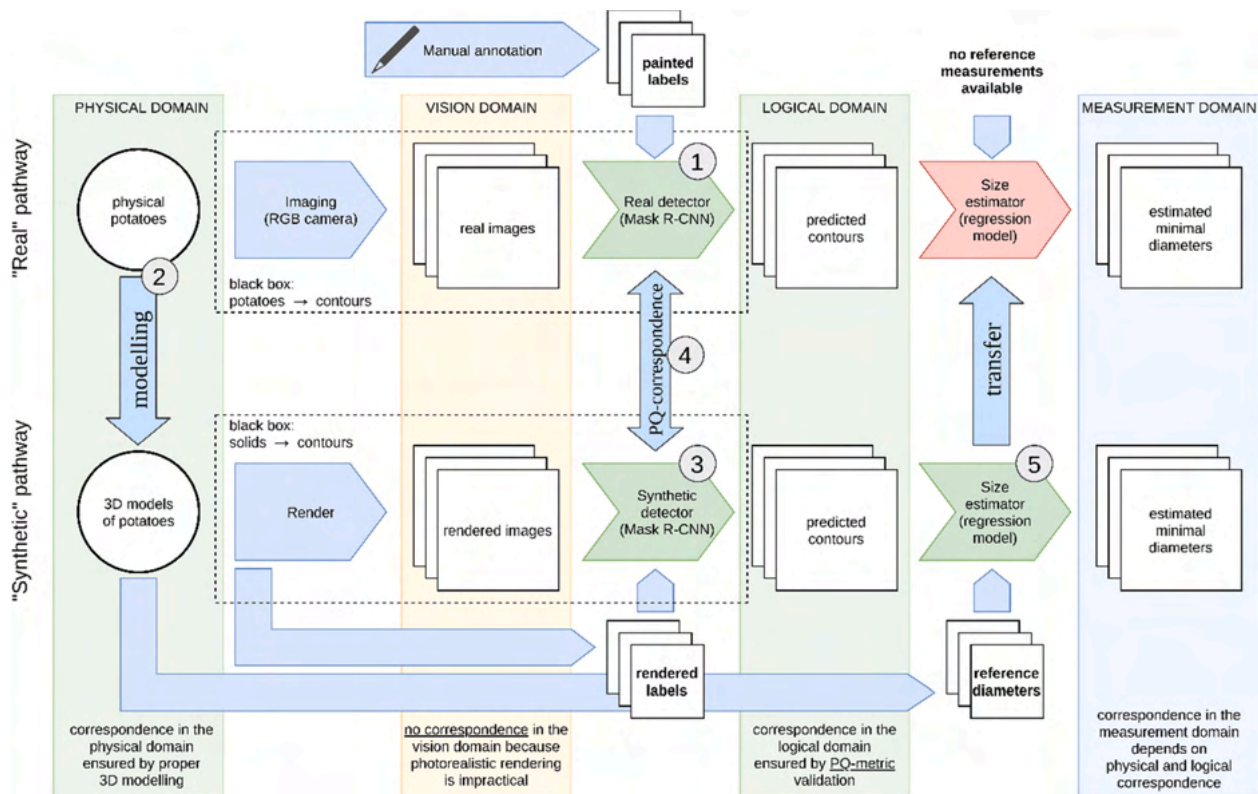


Fig. 6. Map of our proposed approach. Notice how images (in the “vision domain”) are only an intermediate representation.

3. Results

3.1. Instance segmentation

We have performed the first image acquisition in October 2018 in Olawa, Poland. The imaging system has been installed on a Grimme SE-170/60 harvester. The scene was naturally illuminated; the sun light was diffuse due to a rather cloudy sky on the day of acquisition, resulting in soft edges and no hard shadows in the images. Overall, 2589 images in resolution 2304×1728 pixels were acquired. Only a subset of these images was selected to build the dataset that we used in our research: a total of 13 images have received 564 detailed, per-object labels (segmentation contours and bounding boxes). The images were split into training and validation subsets, 8 and 5 images (453 and 111 annotations) respectively. We show an example of a real image and its annotation in Fig. 6.

The synthetic dataset has been rendered in 1792×1344 resolution, which we have chosen arbitrarily in order to ensure similar relative sizes between objects in synthetic and real images. A total of 50 images were rendered in 5 groups, from 17 to 149 objects per image, to simulate different density conditions on the transporter. The process has taken less than 2 h on a computer equipped with an AMD Ryzen 7 3700X CPU and an Nvidia RTX 2070 Super graphics card. The resulting dataset has been split into training and validation subsets, each containing 5 images per group. An example synthetic image and a corresponding annotation has been shown in Fig. 7, compared to a real image.

Both instance segmentation models have been trained in under an hour each on a workstation equipped with an Nvidia RTX 2080 Ti GPU. We do not report the final losses of these models, instead skipping directly to the PQ metrics, giving them in Table 4.

The models displayed a very similar overall quality. The synthetic model made slightly more recognition mistakes, which could be due to some of the synthetic images being generated in a more difficult density conditions, making some of the objects more challenging to detect.

Segmentation Quality is more important, as it defines the accuracy in shape prediction. Both models performed equally well in this domain – the differences are statistically insignificant. Thus, the real and synthetic detection models satisfied the correspondence requirement and no corrective action was necessary. Fig. 7 showcases example predictions of both models on unseen (validation) data.

Next, we analyze the synthetic prediction errors in the function of density. When there are more objects on the scene, the occlusion effect becomes more significant and both recognition and segmentation are more difficult. The bar plot in Fig. 8 displays the quality metrics in the function of density, expressed as the total count of the visible objects. Until about 83 objects per image – which corresponds to approximately 50% of the transporter area being covered with potatoes – the total metric is above 0.90, with recognition score nearly unaffected. When density increases above that level, both of the partial metrics start degrading, leading to a rapid decrease in the panoptic quality. However, increasing density affects the recognition component more significantly than the segmentation: Mask R-CNN does not detect all of the objects when they are heavily crowded, but when it does, it can still segment them relatively well.

3.2. Diameter regression

In order to train the minimal diameter regression model, we have generated a separate synthetic dataset. The same settings have been applied as in the case of instance segmentation dataset, except a constant density factor which we set at 83 objects per image. We have chosen this density because of a satisfactory compromise between the number of objects per image and the recognition rate of the detection model. The exact value is irrelevant at this point of the development, as the diameter regression model is trained on a per-object, as opposed to per-image manner.

In order to simulate different geometries of the imaging system, and thus ensure the regression model does not overfit to one setting, we have



Fig. 7. Comparison between real (left column) and synthetic (right) data. In the top row: example images; middle row: example predictions (with degree of classification confidence in upper left corner of each bounding box); bottom row: ground truth labels.

Table 4
Quality metrics of the instance segmentation models.

Model	Panoptic Quality	Recognition Quality	Segmentation Quality
Real	0.889(± 0.032)	0.976(± 0.018)	0.911(± 0.017)
Synthetic	0.876(± 0.087)	0.948(± 0.060)	0.922(± 0.038)

generated several versions of this dataset with different camera angles. The angles spanned every 15° of a $[15^\circ; 75^\circ]$ range. This resulted in 5 groups of 10 annotated images each. Predictions on this dataset have been acquired using the synthetic detector and then ellipse fitting, and subsequently matched with the object-level annotations. As a result, we created a dataset matching the four ellipse parameters ($\alpha_h, \alpha_w, \beta, \theta$) with the true diameter ($d_{\min}$) of the corresponding 3D object. The 3382 samples in this dataset were then randomly split into training and validation subsets in a 70:30 ratio, with stratification by camera angle.

Training the regression model is very fast even in CPU mode, taking only a few seconds to cover the entire 1500 epochs-iterations (they are

equivalent due to the training batch being equal to the entire dataset), reaching a final L2 loss on the order of 10^{-5} (log-loss = -4.1).

Finally, we validated the model on the remaining subset of the data. Fig. 9 displays scatter plots of predictions against the true values of the minimal diameter, grouped by the camera angle to illustrate potential failures at some angles. We found no such errors, however, except for a small number of outliers belonging to the 30° group. The R^2 metric for all groups varied between 0.829 and 0.890, with the average equal to 0.869. Overall, our model correctly captured the perspective transformation, providing accurate predictions for various camera angles. Not all variance in the data has been accounted for, however. This is naturally due to the estimating nature of the model, which attempts to reconstruct answers in the lack of complete information.

3.3. Experiment in real conditions

To test our proposed solution in real conditions, in December 2019 we have repeated the image acquisition on new material. We were granted access to two big-bags of potatoes of two different sorts, labelled

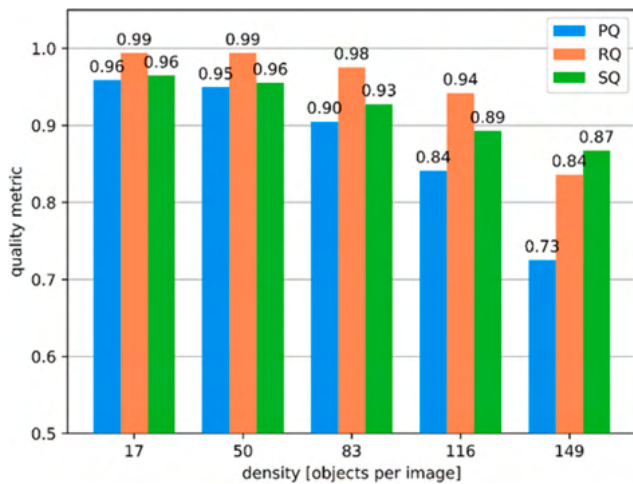


Fig. 8. Panoptic Quality (PQ) scores of the synthetic model, broken down into components – Recognition Quality (RQ) and Segmentation Quality (SQ) – and displayed as a function of object density.

“small” and “large”. Those bags were then poured onto the harvester to emulate true harvest conditions. Other than a general description of the grades, we had no other reference information.

The imaging system was the same as during the first acquisition, although we have not installed it in exact same geometrical configuration. Therefore, calibration with checkerboard pattern was performed again to allow calculation of the viewing angles. Illumination conditions were somewhat different than in October 2018, but the sun light was similarly diffuse. Two series of images were acquired, one for the bag of “small” potatoes, the other for the “large” bag, 18 and 13 images respectively. The images were then undistorted to correct for the radial distortion of the lens. Example images are shown in Fig. 10.

After visually assessing the acquired images, we have found many cases where tubers of very similar sizes appear in both groups. Several examples of this can be seen in the right image, on the center-right as well as lower left (compare with the sizes of objects on the left image).

On the other hand, very few small tubers were present in the batch labelled as “large”. Consequently, we had to assume a larger than anticipated uncertainty in the results, with a certain asymmetry.

We tested the proposed system in two steps. First, we performed instance segmentation using the real model, using it directly with no additional training beyond the original procedure described in 3.1. The model generalizes well and correctly recognized objects in the new images. Problems appeared in the upper ¼ of the image, where the objects were heavily crowded, and Mask R-CNN did not always detect all of them. Also, objects in the bottom section of the image often appeared cropped. Therefore, we only accepted objects located below a threshold of 450 pixels and above 1500 pixels in horizontal dimension (from a total resolution of 1728 pixels). Then, ellipses were fitted to the detected objects, and after conversion to the angular coordinate system, they were fed to the size estimation model. We show the estimation results in Fig. 11 in the form of histograms summarized over each batch.

4. Discussion

Our proposed method allows estimating the minimal diameter of potato tubers from a single image acquired using a consumer-grade RGB camera without any additional illumination. At the core of the method lies an instance segmentation algorithm based on Mask R-CNN. This model can be trained from just a handful of images and still deliver very high accuracy of contour prediction – even when boundaries between neighboring objects are not very clear. As shown in Fig. 6, it is capable of predicting even the occluded parts of the objects, using the contextual information provided in the local curvature of each object together with the learned statistic of the shape of the objects.

Our key contribution is a nonlinear regression model for size estimation, able to estimate the minimal diameter of a tuber from the ellipse fitted to its perceived contour, and a method of training it using simulated data. The model allows estimating this dimension of the object which is not directly observable, basing on the known statistical properties of the shape distribution of the objects. Thanks to a change in coordinate system, our method is not specific to the particular geometry of the imaging system. This coordinate change is done in a simple way, without the need of any additional overhead beyond the ordinary

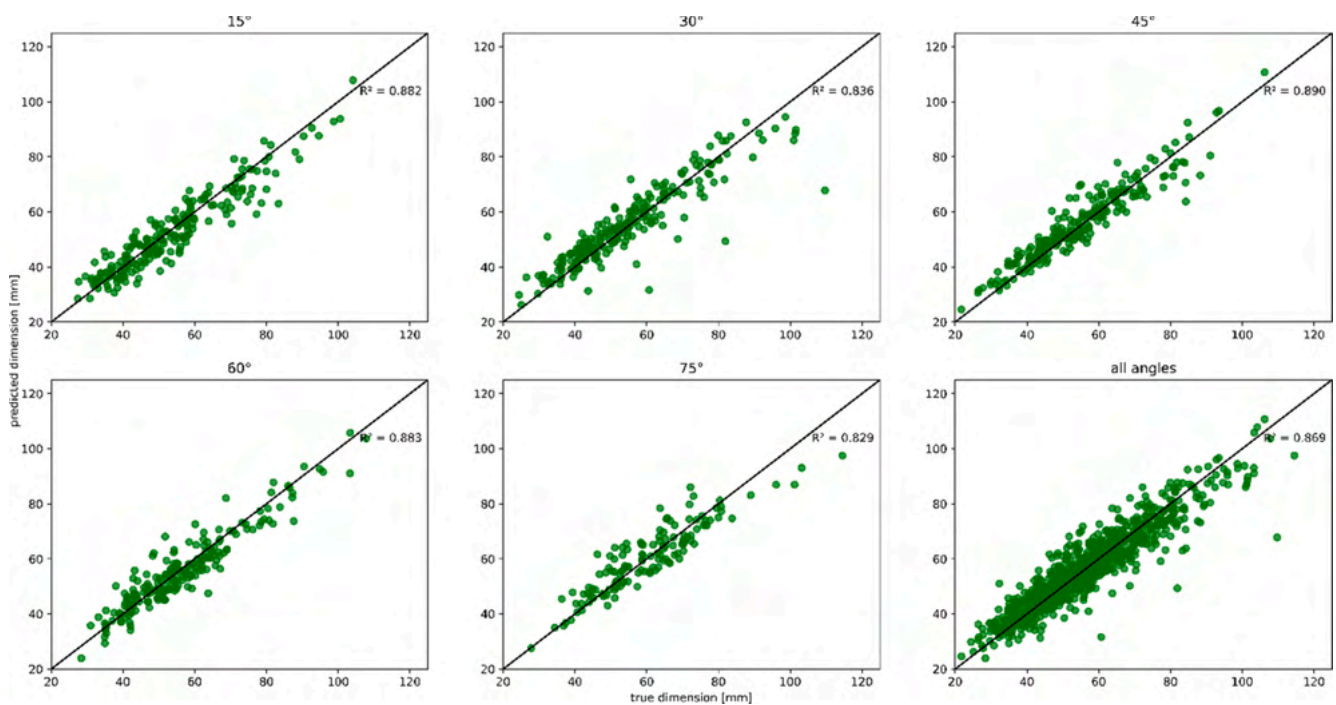


Fig. 9. Analysis of the regression model predictions on the validation subset for different view angles.



Fig. 10. Example images of two different batches of potatoes. Sample from the “large” batch on the left side, sample from the “small” on the right.

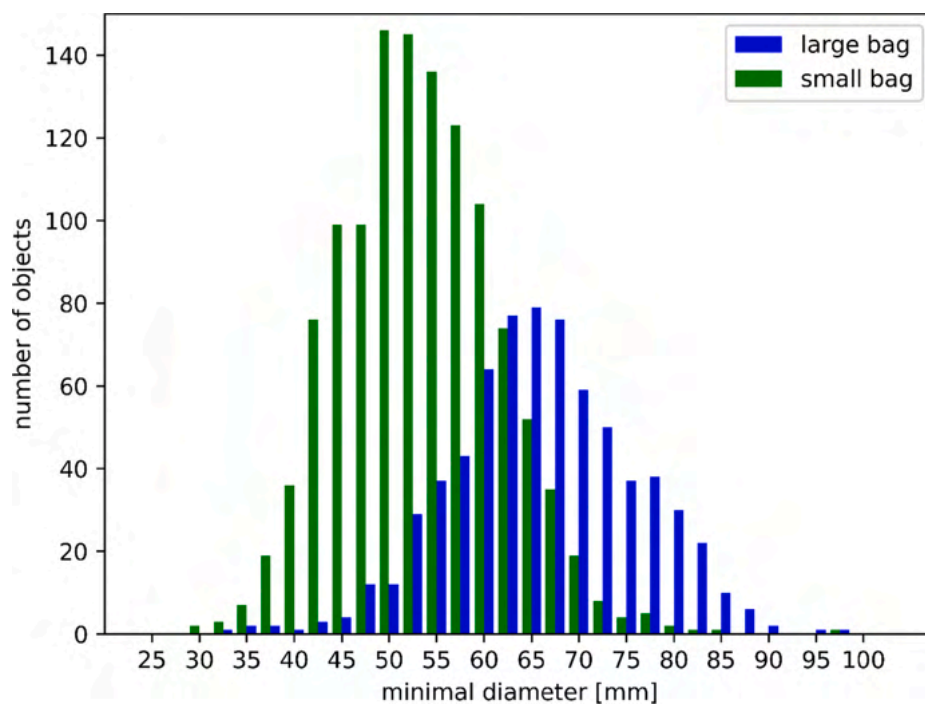


Fig. 11. Histograms of the minimal diameters predicted by our system for each bag of reference potatoes, summed across batches.

camera calibration procedure.

Although our method is generic, several steps must be taken in order to apply it to estimating the sizes of any specific crop. Manual annotation of a dataset is necessary prior to training the instance segmentation model, which is moderately tedious and needs to be done with high precision. Then, in order to calibrate the size estimation regression algorithm, a detailed 3D model of the specific crop must be prepared, following the shape characteristics of that crop. Subsequently, PQ-correspondence between the segmentation models must be ensured. In order to apply a system based on our proposed method in a practical scenario, the limitation of maximum object density must be taken into account. Finally, we stress that we do not propose a method of *measurement* but only of *size estimation*. We build a statistical model to predict a quantity that is not directly visible in the images.

The most important effect that would negatively influence the results would be a difference in shape characteristics between the potato cultivar used to model the synthetic data and the cultivar that is actually grown in the field. Potential improvements might be achieved when using a more accurate potato model, possibly with the spherical-

harmonics deformation model by (Torppa et al., 2007).

Further ways to extend the proposed method could involve improving the quality of rendering, or implementing domain adaptation-based approaches to training of the instance segmentation model. Training a single model that could process both real and synthetic images equally well would circumvent the difficulty in manually annotating the dataset of real images. To improve the detection quality, solution such as Ellipse R-CNN (Dong et al., 2020) could be applied to regress the ellipses directly from the images, instead of fitting them to contours obtained via segmentation with varying accuracy. That approach could be less desired if our method was to be applied to crops that display more regularity than potatoes. If additional geometrical information could be utilized (e.g. location and size of stems in onions), the size estimator could take that into account to improve the predictions. For problems involving even higher degree of inter-object occlusions, amodal segmentation techniques could be applied.

5. Conclusions

We proposed a method for developing detection and size estimation systems for on-line yield monitoring of root crops. Thanks to the use of a state-of-the-art instance segmentation meta-algorithm, Mask R-CNN, the system can detect and segment crops in images acquired using simple, consumer-grade RGB camera with no additional illumination. The segmentation algorithm works correctly even in the case of moderate occlusions between the objects. Our method allows estimating true sizes of the objects even when the images were acquired under significant perspective distortion. It is capable of estimating the minimal diameter of the objects in the case when it is not directly observable in the processed images. This is made possible by learning the shape characteristics of the crop in question from a simulation model.

We showed a specific implementation of our method to estimate sizes of potatoes, first testing the resulting system in simulated conditions, and then deploying it in near-real environment. The system successfully transferred knowledge acquired during simulation-based learning to predict sizes of real crops.

Our method can be applied to other crops (not only root crops) by preparing an annotated dataset for training an instance segmentation model and constructing a simulation environment with a proper three-dimensional model of the desired crop.

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 Piotr Lampa for designing and assembling the image acquisition system. We wish to thank Piotr Lampa and Emil Nowosielski for their work on the image acquisition process itself. We also wish to extend our thanks to Aleksy Kubik and Klaudiusz Matejka Farm, who supported the work by hosting the field experiments and providing the potatoes.

This research was supported by the National Centre for Research and Development (BIOSTRATEG3/343547/8/NCBR/2017).

References

- Bargoti, S., Underwood, J.P., 2017. Image segmentation for fruit detection and yield estimation in apple orchards. *J. Field Robot.* 34 (6), 1039–1060. <https://doi.org/10.1002/rob.21699>.
- Barth, R., Jsselmuiden, J., Hemming, J., Van Henten, E.J., 2018. Data synthesis methods for semantic segmentation in agriculture: a capsicum annum dataset. *Comput. Electron. Agric.* 144, 284–296. <https://doi.org/10.1016/j.compag.2017.12.001>.
- Blender Online Community, 2019. Blender – A 3D Modelling and Rendering Package. Stichting Blender Foundation, The Blender Foundation, Amsterdam <http://www.blender.org>.
- Boatswain Jacques, A.A., Adamchuk, V.I., Cloutier, G., Clark, J.J., Miller, C., 2018. Development of a machine vision yield monitor for shallot onion harvesters. In: Proceedings of the 14th International Conference on Precision Agriculture. Montreal, Quebec, Canada. https://ispag.org/proceedings/?action=author_abstracts.
- Chen, Steven W., Shivakumar, Shreyas S., Dcunha, Sandeep, Das, Jnaneshwar, Okon, Edidiong, Chao, Qu, Taylor, Camillo J., Kumar, Vijay, 2017. Counting apples and oranges with deep learning: a data-driven approach. *IEEE Robot. Automat. Lett.* 2 (2), 781–788. <https://doi.org/10.1109/LRA.2017.2651944>.
- Chinchuluun, R., Lee, W.S., Ehsani, R., 2009. Machine vision system for determining citrus count and size on a canopy shake and catch harvester. *Appl. Eng. Agric.* 25 (4), 451–458.
- Dong, Wenbo, Roy, Pravakar, Peng, Cheng, Isler, Volkan, Ellipse R-CNN: Learning to Infer Elliptical Object from Clustering and Occlusion, ArXiv:2001.11584 [Cs], January 30, 2020. <http://arxiv.org/abs/2001.11584> (preprint).
- ElMasry, Gamal, Cubero, Sergio, Moltó, Enrique, Blasco, José, 2012. In-line sorting of irregular potatoes by using automated computer-based machine vision system. *J. Food Eng.* 112 (1), 60–68. <https://doi.org/10.1016/j.jfoodeng.2012.03.027>.
- Gongal, A., Amata, S., Karkee, M., Zhang, Q., Lewis, K., 2015. Sensors and systems for fruit detection and localization: a review. *Comput. Electron. Agricul.* 116, 8–19. <https://doi.org/10.1016/j.compag.2015.05.021>.
- Habaragamuwa, Harshana, Ogawa, Yuichi, Suzuki, Tetsuhito, Shiigi, Tomoo, Ono, Masanori, Kondo, Naoshi, 2018. Detecting greenhouse strawberries (mature and immature), using deep convolutional neural network. *Eng. Agric. Environ. Food* 11 (3), 127–138. <https://doi.org/10.1016/j.eaef.2018.03.001>.
- He, Kaiming, Gkioxari, Georgia, Dollár, Piotr, Girshick, Ross, 2017. Mask R-CNN, 2961–69, 2017. https://openaccess.thecvf.com/content_iccv_2017/html/He_Mask_R-CNN_ICCV_2017_paper.html.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>.
- Hofstee, J.W., Molema, G.J., 2003. Volume estimation of potatoes partly covered with dirt tare. In: 2003, Las Vegas, NV July 27–30, 2003. American Society of Agricultural and Biological Engineer. <https://doi.org/10.13031/2013.15380>.
- Kestur, Ramesh, Meduri, Avadesh, Narasipura, Omkar, 2019. MangoNet: a deep semantic segmentation architecture for a method to detect and count mangoes in an open orchard. *Eng. Appl. Artificial Intell.* 77, 59–69. <https://doi.org/10.1016/j.engappai.2018.09.011>.
- Khojastehnazhand, Mostafa, Mohammadi, Vahid, Minaei, Saied, 2019. Maturity detection and volume estimation of apricot using image processing technique. *Scientia Horticulturae* 251, 247–251. <https://doi.org/10.1016/j.scientia.2019.03.033>.
- Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P., 2019. Panoptic segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9396–9405. <https://doi.org/10.1109/CVPR.2019.00963>.
- Kusumam, K., Krajník, T., Pearson, S., Duckett, T., Cielniak, G., 2017. 3D-vision based detection, localization, and sizing of broccoli heads in the field. *J. Field Robot.* 34 (8), 1505–1518. <https://doi.org/10.1002/rob.21726>.
- Lee, Young-Joo, Kim, Ki-Duck, Lee, Hyeon-Seung, Shin, Beom-Soo, 2018. Vision-based potato detection and counting system for yield monitoring. *J. Biosyst. Eng.* 43 (2), 103–109. <https://doi.org/10.5307/JBE.2018.43.2.103>.
- Lin, Guichao, Tang, Yunchao, Zou, Xiangjun, Cheng, Jiabing, Xiong, Juntao, 2020. Fruit detection in natural environment using partial shape matching and probabilistic hough transform. *Precis. Agric.* 21 (1), 160–177. <https://doi.org/10.1007/s11119-019-09662-w>.
- Long, Yaowei, Wang, Yuhao, Zhai, Ziming, Wu, Li, Li, Minzan, Sun, Hong, Su, Qinghua, 2018. Potato volume measurement based on RGB-D camera. In: IFAC-PapersOnLine, 6th IFAC Conference on Bio-Robotics BIORBOTICS 2018, vol. 51, no. 17, Springer, pp. 515–520. <https://doi.org/10.1016/j.ifacol.2018.08.157>.
- Mavridou, E., Vrochidou, E., Papakostas, G.A., Pachidis, T., Kaburlasos, V.G., 2019. Machine vision systems in precision agriculture for crop farming. *J. Imag.* 5 (12), 89. <https://doi.org/10.3390/jimaging5120089>.
- Momin, M.A., Rahman, M.T., Sultana, M.S., Igathinathane, C., Ziauddin, A.T.M., Grift, T. E., 2017. Geometry-based mass grading of mango fruits using image processing. *Inform. Process. Agricul.* 4 (2), 150–160. <https://doi.org/10.1016/j.inpa.2017.03.003>.
- Noordam, Jacco C., Otten, Gerwoud W., Timmermans, Toine J.M., van Zwol, Bauke H., 2000. High-speed potato grading and quality inspection based on a color vision system. In: Machine Vision Applications in Industrial Inspection VIII, vol. 3966. International Society for Optics and Photonics, pp. 206–217. <https://doi.org/10.1117/12.380075>.
- Pandey, Nikhil, Kumar, Suraj, Pandey, Raksha, 2019. Grading and defect detection in potatoes using deep learning. In: Communication, Networks and Computing, edited by Shekhar Verma, Ranjeet Singh Tomar, Brijesh Kumar Chaurasia, Vrijendra Singh, and Jemal Abawajy. Communications in Computer and Information Science. Springer, Singapore, pp. 329–339. https://doi.org/10.1007/978-981-13-2372-0_29.
- Paszke, Adam, Gross, Sam, Chintala, Soumith, Chanan, Gregory, Yang, Edward, DeVito, Zachary, Lin, Zeming, 2017. Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic Differentiation in PyTorch. In NIPS-W.
- Razmjoo, Navid, Somayeh Mousavi, B., Soleymani, F., 2012. A real-time mathematical computer method for potato inspection using machine vision. *Comput. Math. Appl.* 63 (1), 268–279. <https://doi.org/10.1016/j.camwa.2011.11.019>.
- Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., McCool, C., 2016. DeepFruits: a fruit detection system using deep neural networks. *Sensors* 16 (8), 1222. <https://doi.org/10.3390/s16081222>.
- Si, Yongsheng, Sindhuja Sankaran, N., Knowles, Richard, Pavek, Mark J., 2018. Image-based automated potato tuber shape evaluation. *J. Food Measur. Charact.* 12 (2), 702–709. <https://doi.org/10.1007/s11694-017-9683-2>.
- Su, Qinghua, Kondo, Naoshi, Li, Minzan, Sun, Hong, Riza, Dimas Firmanda Al, Habaragamuwa, Harshana, 2018. Potato quality grading based on machine vision and 3D shape analysis. *Comput. Electron. Agric.* 152, 261–268. <https://doi.org/10.1016/j.compag.2018.07.012>.
- Szczypiński, Piotr M., Strzelecki, Michał, Materka, Andrzej, Klepaczk, Artur, 2009. MaZda—a software package for image texture analysis. *Comput. Methods Prog. Biomed.* 94 (1), 66–76. <https://doi.org/10.1016/j.cmpb.2008.08.005>.
- Torppa, Johanna, Valkonen, Jari P.T., Muinonen, Karri, 2006. Three-dimensional stochastic shape modelling for potato tubers. *Potato Res.* 49 (2), 109–118. <https://doi.org/10.1007/s11540-006-9010-5>.
- TrueSort, Ellips B.V., Eindhoven, Netherlands, 2020. Available at: <https://ellips.com/fruit-and-vegetables/potato-sorting/>.
- Wang, C., Huang, W., Zhang, B., Yang, J., Qian, M., Fan, S., Chen, L., 2016. Design and implementation of an automatic grading system of diced potatoes based on machine vision. *IFIP Adv. Inform. Commun. Technol.* 479, 202–216. https://doi.org/10.1007/978-3-319-48354-2_22.
- Wu, Yuxin, Kirillov, Alexander, Massa, Francisco, Lo, Wan-Yen, Girshick, Ross. Detectron2, 2019. <https://github.com/facebookresearch/detectron2>.

- Yu, Chuang, Fan, Xiang, Zhuhua, Hu, Xia, Xin, Zhao, Yaochi, Li, Ruqing, Bai, Yong, 2020. Segmentation and measurement scheme for fish morphological features based on mask R-CNN. Inform. Process. Agric. <https://doi.org/10.1016/j.inpa.2020.01.002>.
- Yu, Yang, Zhang, Kailiang, Yang, Li, Zhang, Dongxing, 2019. Fruit detection for strawberry harvesting robot in non-structural environment based on mask-RCNN. Comput. Electron. Agric. 163 <https://doi.org/10.1016/j.compag.2019.06.001>.
- Zhang, H., Li, Z., Liu, L.i., Liang, Z., 2018. Geometry-based mass grading of betelnut using image processing. In: 2018 13th World Congress on Intelligent Control and Automation (WCICA), pp. 1014–1019. <https://doi.org/10.1109/WCICA.2018.8630602>.

(original submission)

Digital Object Identifier

Amodal instance segmentation for mealworm growth monitoring using synthetic training images

PRZEMYSŁAW DOLATA¹, PAWEŁ MAJEWSKI², PIOTR LAMPA³, MACIEJ ZIĘBA^{1,4}, JACEK REINER³¹Department of Artificial Intelligence, Wrocław University of Science and Technology, 50-372 Wrocław, Poland²Faculty of Information and Communication Technology, Wrocław University of Science and Technology, 50-370 Wrocław, Poland³Faculty of Mechanical Engineering, Wrocław University of Science and Technology, 50-370 Wrocław, Poland⁴Tooploox, 53-601 Wrocław, Poland

Corresponding author: Przemysław Dolata (e-mail: przemyslaw.dolata@pwr.edu.pl).

ABSTRACT Automatic dimensioning of mealworms based on computer vision is challenging due to occlusions. Amodal instance segmentation (AIS) could be a viable solution, but the acquisition of annotated training data is difficult and time-consuming.

This work proposes a new method to prepare data for training AIS models that reduces the human annotation effort significantly. Instead of acquiring the occluded images directly, only images of fully visible larvae are acquired and processed, allowing obtaining their contours via automatic segmentation. Next, synthetic images with occlusions are generated from the database of automatically extracted instances. The generation procedure uses simple computer graphics tools and is computationally inexpensive, yet yields images that allow training off-the-shelf AIS models. Since those models need to be tested on real data, which requires manual annotation, a data acquisition method that significantly simplifies the test set annotation process is demonstrated.

Results are reported in terms of the amodal segmentation quality as well as the accuracy of larvae dimensioning, measured using the histogram intersection metric. The best-performing model achieves a mean average precision of 0.41 and a histogram intersection of 0.77, confirming the effectiveness of the proposed method of data acquisition and generation. The method is not specific to mealworm detection and could be applied to other similar problems where object occlusions pose a challenge.

INDEX TERMS amodal segmentation, instance segmentation, dense scenes, dataset synthesis, synthetic images, *Tenebrio molitor*

I. INTRODUCTION

Large-scale rearing of insects for feed and food - mainly *Hermetia illucens* and *Tenebrio molitor* (mealworm) - is becoming increasingly popular in Europe [1], and researchers are developing machine vision systems to help farmers with rearing monitoring and feeding control [2]. Automatic phenotyping, understood here as measuring geometric traits of the larvae, allows obtaining their size distribution, which is an important indicator that characterizes the population status. Object detection and instance segmentation methods can be used to detect individual larvae and obtain their contours, allowing measurement. But in practice, images of mealworm rearing boxes are very densely populated, with the larvae frequently occluding one another. This makes detection challenging and restricts typical segmentation methods to the prediction of

visible regions of each instance. Visible parts of the larvae are not fully representative of their actual physical characteristics, hence limiting measurement to those parts would yield inaccurate results. For this reason, this work proposes to formulate the problem of detecting and segmenting larvae as an amodal instance segmentation task.

Instance segmentation is a computer vision task that unifies object detection and segmentation: for each object present in the scene, an individual segmentation mask is expected as the output. *Amodal* instance segmentation (AIS) is a subtype of the instance segmentation task, in which the expected result is a segmentation mask of the object even when said object is not fully visible. The idea draws from the phenomenon observed in psychology, *amodal perception*: the ability of humans to perceive incompletely observed objects

as a complete whole. In the AIS setting, scenes of interest exhibit frequent occlusions, i.e. objects overlap one another, resulting in some of them being only partially observable. This observable component is called a *modal* part. AIS asks to predict the *amodal* part, which is the complete extent of the object, including both the visible and the occluded regions.

Training AIS models requires a large number of images with amodal annotation. Obtaining this type of ground truth for very dense, heavily occluded scenes is exceptionally expensive and difficult. High object count and proximity make the task of human annotators cumbersome, requiring significantly more focus than is the case with modal segmentation, and thus more time spent on annotating a single image. Due to the inherent uncertainty in annotating occluded regions, annotators have to make subjective decisions when constructing the labels, resulting in frequent mistakes. This difficulty remains a considerable obstacle in the adoption of AIS in industrially applied computer vision. For this reason, many approaches to AIS emphasize methodologies for synthesizing training data over improving segmentation models. They sometimes employ complex generative models that are expensive to train and do not fully alleviate the problem of manual amodal annotation.

This work aims to solve the problem of obtaining ground truth for the training of AIS models dedicated to engineering applications, in particular to the monitoring of the growth of mealworm larvae by determining the size distribution of larvae in the population in the form of length histograms. The proposed approach is to reduce the problem of amodal data acquisition to the much easier modal data acquisition and leverage computer graphics techniques in order to generate amodal data. Manually annotating thousands of occluded larvae in the rearing boxes is expensive, but the similar *modal* task of segmenting fully visible larvae on a contrasting background could be fully automated. A database of individual, pre-segmented larva instances, together with a database of natural backgrounds, would allow the synthesizing of perfectly annotated images with arbitrary degrees of density and occlusion using basic computer graphics tools. This would be much simpler and substantially less computationally intensive than using sophisticated generative models, yet still allow directly training AIS models.

To summarize, the following contributions are made in this paper:

- a novel method for the generation of annotated training data for amodal instance segmentation,
- a methodology for the acquisition of the real test data, allowing objective amodal annotation,
- a comprehensive analysis of the amodal instance segmentation models, trained using the proposed method, on the task of mealworm growth monitoring.

II. RELATED WORKS

A. AMODAL INSTANCE SEGMENTATION METHODS

The phenomena of amodal perception and amodal completion have long been the subject of study by psychologists (see e.g.

[3]). The first to consider amodal perception in the context of computer vision were [4], who introduced the term “Amodal Instance Segmentation” (AIS). [5] extended the concept onto the semantic segmentation task, introducing *semantic* amodal segmentation and creating a dedicated dataset, a subset of the COCO dataset ([6]) with extra amodal annotations, known as COCOA. [7] proposed a generative approach for simultaneous segmentation and inpainting of occluded objects, SeGAN. They trained their model on photorealistic synthetic images and evaluated it on real images, proving that such a direct domain transfer is possible. [8] presented the first end-to-end trainable approach for amodal instance segmentation, ORCNN, extending Mask R-CNN ([9]) with separate prediction heads for modal and amodal segmentation. They demonstrated that augmenting modal training data might be sufficient for achieving decent performance on the amodal test split of their novel amodal dataset, D2SA. The idea of separating modal and amodal predictors was then researched by [10], who found that utilizing relationships between those two components can lead to a significant improvement in amodal segmentation quality.

Another key dataset for amodal segmentation, KINS, was proposed by [11] along with a novel amodal segmentation network using a technique dubbed Multi-Level Coding. [12] highlighted the importance of depth ordering and introduced a new type of representation (a semantics-aware distance map), allowing their amodal segmentation model to predict the occluded boundaries as well as order the predicted instances. [13] argued that the amodal perception in humans is not based on the visual stimulus alone, but uses prior knowledge about the shapes of objects. Their model directly encapsulates this notion, learning shape priors and using them to drive amodal segmentation. The idea of a prior was also explored by [14], who used Bayesian models, training them on modal data and considering amodal segmentation as an out-of-distribution generalization. [15] modeled amodal instance segmentation as two separate tasks: the occluded object segmentation, and the occluding object segmentation, training a single model, BCNet, to exploit relationships between objects. This concept was significantly extended by [16] who used a transformer-based extension to DETR ([17]) to model relationships between the visible part of the occluded object, its invisible part, and the occluding object.

The amodal perception was studied in other areas of computer vision as well. Specialized methods were proposed for segmenting objects in video streams, such as [18], [19], [20]. Amodal panoptic segmentation, introduced by [21], seeks to simultaneously segment regions of amorphous “stuff” classes and individual objects, both their visible and invisible parts. Another related task is image inpainting, where reconstruction from incomplete information is requested in image space as opposed to label space - see [22] for the survey on the subject.

B. TRAINING DATA GENERATION FOR AMODAL SEGMENTATION

Occlusions and the need to predict complete boundaries of the partially observed objects appear increasingly often in industrial applications of computer vision. One of the first mentions of amodal completion in the context of computer vision appears in [23], where bounding boxes of occluded objects were completed to predict object sizes. The problem of acquiring the ground truth data has already arisen. In order to generate the amodal labels for the PASCAL dataset used in that research, the authors relied on the PASCAL 3D+ dataset, which contains approximate 3D models aligned to objects in the PASCAL VOC images. However, this was only possible due to the use of a specific benchmark dataset, one that had already been well-researched, and happened to have a sibling dataset with the 3D models, which allowed the amodal information to be constructed. In applications such as manufacturing, robotics, or agriculture, where available annotated data is scarce, preparation of *amodal* annotations is not as straightforward. This makes the adoption of amodal instance segmentation methods for problems in those areas very challenging.

The dominant approach for preparing data for the training of amodal segmentation models in industrial applications is still to acquire raw images and then perform some form of manual annotation in the amodal domain. For example, [24] proposed a novel network architecture for amodal segmentation of overlapping cane sugar crystals. They trained their model on manually annotated data, highlighting the need for an experienced worker to oversee the annotation process, as well as for a strict protocol to be followed to obtain consistent annotations. A similar approach was followed by [25] in the context of remote sensing. Image data (in this case acquired by aerial imaging) was annotated by specialists using dedicated software. Each annotator labeled a different class of objects, ensuring independence of their predictions, inherently uncertain due to incomplete information. In [26], an image-based food calorie estimation system is proposed, with a Mask R-CNN model for amodal instance segmentation of foodstuffs, and a separate model for estimation of their weights. Training images had manually constructed amodal annotations, although the details of the annotation procedure were not given. [27] faced a conceptually similar problem of size estimation: in their case, a dedicated amodal segmentation network was proposed for the problem of gravel particle sizing. It was trained on a manually annotated dataset, created by following a specific protocol. Their approach is interesting because not only does it demonstrate an application of amodal instance segmentation, but also reports performance in a problem-specific (downstream) metric, beyond the generic mAP.

The common factor of all research mentioned so far is the process of manual annotation relying on expert knowledge. In all cases, the annotators were faced with images containing occluded objects and asked to create amodal annotations, with the assumption that an expert should be able to reconstruct the

complete extent of the object boundary. It is not unreasonable to question the objectivity of this approach - after all, in this setting, the expert does not have access to the true shape of the object; they merely infer it using their domain knowledge. The resulting annotation is thus only an estimation, as the actual true shape of the object - the *ground truth* - remains unknown. For this reason [28], in order to detect and segment occluded broccoli heads in field conditions, propose a data acquisition procedure that gives the annotators access to the actual ground truth. In their work, the objects are first imaged in full view, and then with artificially introduced occlusions. This allows annotating each object when it is fully visible and then transferring this annotation to the occluded images.

Several applications of amodal segmentation have been presented in the context of RGB-D imaging. The inclusion of depth channel allowed [29] to automatically generate the amodal annotations for apple fruits by assuming their geometry is spherical. Their approach relied on fitting a sphere to each partially occluded object, and then projecting it onto the image plane. In the context of autonomous driving, [30] trained an amodal segmentation model for shape and pose estimation from an incomplete LiDAR sensor signal. Their training data was artificially generated from vehicle meshes in the form of CAD files. [31] demonstrate amodal 3D reconstruction for robotic manipulation, learning from an artificial dataset using 3D simulation and OpenGL-based rendering, building their scenes from the ShapeNet database of 3D shapes. [32] follow a conceptually similar pipeline for data generation, but render sequences of images instead, since their proposed pipeline includes a motion prediction component. [33] implement a photo-realistic rendering pipeline and fit their model with a Sim2Real transfer component to further bridge the gap between the rendered training images, and the real test images. In summary, 3D modeling can be a convenient way to generate data for learning amodal segmentation, but only if objects in question can be approximated with primitives, or a database of reference 3D shapes is available. For problems concerning atypical, irregular objects, this approach may not be equally applicable. Additionally, the usage of RGB-D imagery also fundamentally limits the transfer of proposed methodologies, as it places a basic constraint over the application domain in the form of a non-standard and costly sensor.

The idea of generating training data by simulating occlusions has been applied in the two-dimensional modality as well, for example in [34] for amodal segmentation of cucumbers. In this work, images of fully visible cucumbers were acquired and annotated, and then random rectangles were replaced with random non-object texture, with the annotation preserved. A more advanced approach was proposed in [35] for amodal recovery of human silhouettes. They also began with a non-occluded dataset with modal annotations, but generated the occlusion cases by randomly pasting other objects as foreground. This allowed the creation of realistic-looking occlusions - as opposed to rectangular cutouts of [34] - but only if the occluding object was of a different

class. A similar idea was adopted in [36] for the problem of amodal instance segmentation of piglets in farrowing pens. The authors augmented their manually annotated dataset with random occlusions created by pasting random clutter objects into the image foreground. In those works, the objects of interest are artificially occluded with some “clutter” objects, and occlusions between objects of the same class are not considered.

The most sophisticated data generation pipeline was proposed by [37] for the problem of size estimation of rock and gravel particles. First, reference images of particles were rendered from a database of 3D models. Those were used to train a Diffusion-GAN model, which was subsequently used to generate an even larger particle dataset. Next, a purpose-built algorithm was used to generate cluttered scenes containing multiple particles sampled from this dataset. Finally, the images generated this way were post-processed with a CycleGAN model to improve their visual similarity to the real (photographic) images. While each step of the proposed pipeline was found to improve the segmentation accuracy of the trained model, it is computationally expensive and complex. Additionally, it requires an initial database of 3D models and textures, which may not be easily available for other tasks.

In summary, amodal segmentation techniques are slowly being adopted in various problem domains, offering performance improvements not only in the pure segmentation metrics such as mAP but most importantly in the problem-specific metrics, for example lowering the mean squared error of dimension estimation. Most of the related works dedicate considerable attention to the problem of acquiring training data. Manual annotation is more expensive, time-consuming, and error-prone in the amodal domain due to the inherent uncertainty in the images, requiring an expert annotator to infer the contour of the occluded region of each object. Many ideas for automating this process appear, starting with introducing synthetic occlusions. Those methods have considerable limitations: they are only demonstrated in the simple case of an object of interest being occluded by non-interesting clutter; it is unclear how they extend to occlusions between multiple objects of interest. Furthermore, they still require some form of annotation (most often manual) of the initial, modal dataset. This issue can be mitigated to some extent using simulated data. In particular, in 3D contexts such as robotics, arbitrary datasets can be rendered from reference meshes. Those references may not be so readily available in the case of highly irregular objects, which are common for example in agriculture. Additionally, relying on expensive RGB-D sensors to facilitate the segmentation severely limits the practical applicability of the developed methods. Finally, the rise of generative modeling gives hope that entire datasets for amodal instance segmentation could be generated using a properly trained model. Unfortunately, so far those approaches are not only very complicated and computationally expensive but also still require some initial data for training themselves.

Compared to the reference approaches, the method for generating artificial data proposed in this research is conceptually simple, as it does not require high-fidelity 3D rendering software. It can be applied in a typical machine vision setting, without depending on costly RGB-D sensors. Furthermore, it does not require sophisticated generative models, which makes it significantly less computationally expensive. Very few assumptions are placed on the objects in question, enabling modeling of occlusions between objects of the same class. Finally, the method requires very little human input on the annotation stage, providing objective ground truth without relying on other sources such as 3D models or textures.

C. MEALWORM GROWTH MONITORING

The problem of monitoring the growth of *T. molitor* larvae has already been addressed in the literature. In [38], researchers proposed a method based on classical image processing methods, which consisted of extracting larvae segments from images, classifying the segments, and calculating their size. As a limitation of this approach should be considered the possible difficulty of interpretation of the results by the farmer and the questionable robustness of the approach with the changing appearance of larvae during rearing, e.g. a poorly visible segment for small larvae.

In [2], a method for sizing larvae based on the obtained binary masks after segmentation using Mask R-CNN was proposed. Only non-occluded larvae were used to determine larval length distribution, and the determination of non-occlusion was based on the extracted and classified larval segments. A larva received a non-occluded label when the end segments represented the head and tail. The limitations of this method were the exclusion of occluded larvae when phenotyping the larvae population and the relatively long processing time.

[39] proposed an end-to-end solution for the problem of larvae dimensioning and determining size distribution statistics. The researchers developed a regression convolutional neural network (RegCNN) to calculate quartiles (lower, median, upper) of larval widths, and quartiles for larval length and volume were determined indirectly using determined linear regression models length(width) and volume(width). The work demonstrated the robustness of the proposed method to dense scenes. The knowledge transfer mechanism significantly reduced sample labeling time - a few manually labeled samples were used only for calibration purposes. The paper also pointed out the importance of future work in the area of amodal segmentation and the study of the effect of larval density on the obtained results.

It is, therefore, clear that the issue of occlusions is among the challenging factors in monitoring of mealworm growth. Overcoming this problem might be possible via manual labor (e.g. a quality assurance employee manually sampling and measuring the larvae) or modifications of the domain hardware (e.g. a robotic solution to automate the sampling and measurement), but those approaches are necessarily slow and costly, perhaps prohibitively so. Imaging and estimation

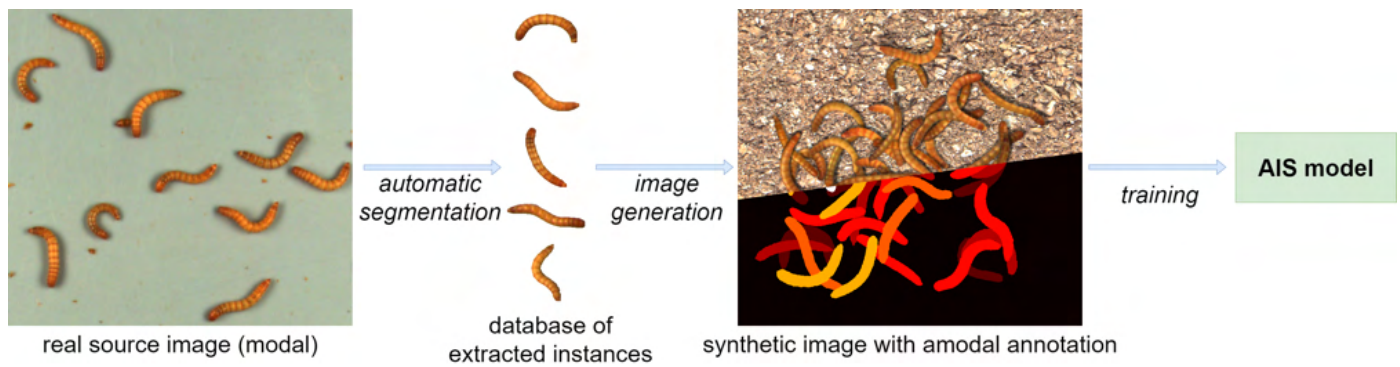


FIGURE 1. Proposed concept for the training pipeline. Instances are extracted from real source images via automatic segmentation. Synthetic images are generated using a database of instances. Amodal annotation is constructed in parallel (heatmap colors represent visibility ordering). AIS models are trained on the resulting dataset.

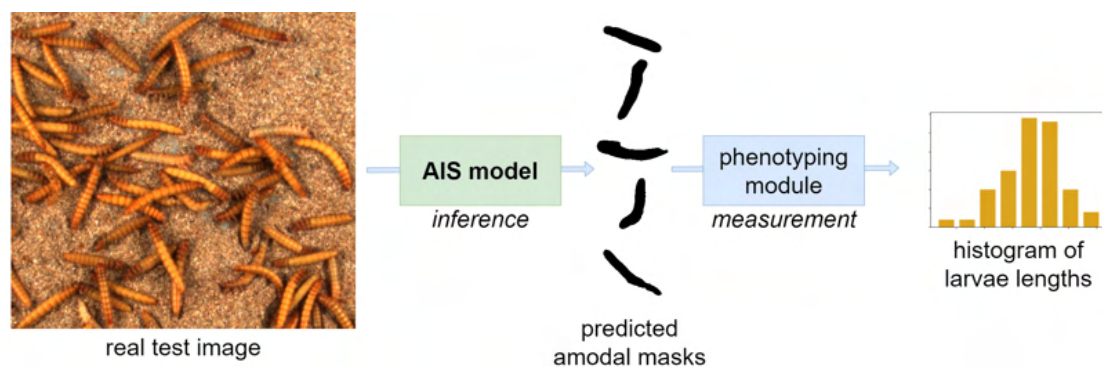


FIGURE 2. Proposed concept for inference. An AIS model trained on synthetic images is used to detect and segment instances on real test images. Predicted amodal instance masks are passed to the phenotyping module that measures the lengths of the individual larvae and outputs a histogram.

based on amodal segmentation have the potential for accurate remote sensing and prediction at a fraction of the cost and without extra time overhead. However, due to the difficulty in obtaining the training data, it is currently not possible to apply off-the-shelf amodal instance segmentation models to this problem.

III. MATERIALS AND METHODS

A. OVERVIEW

In order to overcome the problem of the acquisition of annotated training data for AIS training, this work proposes an alternative pipeline that reduces the process of obtaining ground truth labels to the modal domain and relies on simple computer graphics tools to generate amodal data. The first step, discussed in section III-B, is to obtain images of instances in simplified conditions: occlusion-free and easy to extract from the background. Subsequently, the set of instances is extracted from the source images. Note that due to a fully modal characteristic of the acquired images, perfect information about the contour of each instance is available. Finally, described in section III-C, computer graphics tools are used to combine a randomly sampled subset of instances into a set of new, synthetic images. Since the shape of each of the instances was known during synthesis, amodal labels can be automatically constructed for each synthetic image. Fig. 1

illustrates the proposed training pipeline.

The dataset of images obtained in this way allows the training of AIS models directly. The inference scheme, shown in Fig. 2, begins with using the trained model to detect instances and their amodal masks. Then, predictions are passed to a dedicated phenotyping module that performs relevant measurements - in this case, lengths of detected larvae. The final output of the algorithm is a histogram of lengths that can be compared with the ground truth histogram.

Several state-of-the-art models are trained and compared in the course of this work, details are given in section III-D. Performance evaluation, outlined in section III-E, is conducted on two levels: first, the quality of object detection and amodal segmentation is compared across the models, and then the impact of segmentation quality on the measurement quality is examined. However, to assess the actual performance of the phenotyping system, the evaluation cannot be performed on synthetic images. For this purpose, a separate test set of real images must be obtained, along with amodal ground truth labels. “Real” in this context shall mean images acquired with the physical acquisition system, on a physical scene - as opposed to synthetically generated images. Since annotating images with intense occlusions is challenging, a procedure to acquire and annotate the test set efficiently is proposed in section III-B3.

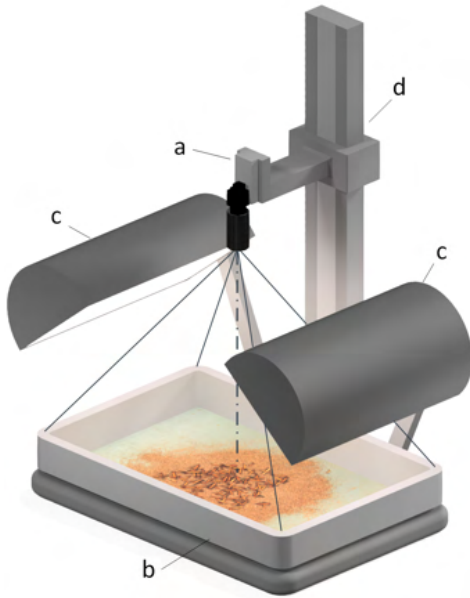


FIGURE 3. Test stand for image acquisition, a) camera with lens, b) rearing box with samples, c) illuminators, d) camera mount.

B. DATA ACQUISITION

1) Image acquisition

All the images used in this study were collected on a prepared test stand (Fig. 3). It used a color camera Phoenix PHX120S-CC (LUCID Vision Labs, Richmond, Canada), with a resolution of 4096 x 3000 pixels and a 12 mm lens. The camera was installed at a distance of 490 mm above a rearing box into which the insect and medium samples were placed. The field of view was illuminated by diffusing reflectors with fluorescent lamps. This was done to minimize light reflections and sharp shadows.

The raw images suffered from uneven brightness due to insufficient illumination of some regions of the rearing box and distortion, which is a drawback of the optical system used. Uneven image brightness was compensated using an underexposure map derived from the gray pattern image. The undistortion process was performed using a chessboard pattern [40]. The described method of acquisition and pre-processing was common to both training and test images.

2) Training dataset

a: Background acquisition

In order to generate synthetic images closely resembling the real ones, a set of real backgrounds corresponding to the texture of empty (i.e. containing no larvae) rearing boxes must be acquired first. The first iteration of the data acquisition procedure is thus executed on scenes containing boxes filled with sand and other matter commonly present during the growth of the larvae. Several different background images are acquired, maximizing variety during the generation procedure. All of those images are acquired with identical settings of the acquisition system, ensuring image resolution consistency throughout the dataset.

b: Instance acquisition

Acquisition of images containing fully visible, easy-to-extract larvae is the crucial step of the proposed data preparation pipeline. Larvae are placed into a completely empty gray-blue rearing box - the color was chosen to highly contrast with the natural hue of the larvae. The purpose is to make automatic segmentation of the larvae easier. Additionally, the placement of the larvae is deliberately sparse, so as to minimize the probability of the instances occluding one another. After capturing the image, larvae are removed and the box is cleaned before the next iteration, ensuring that no larvae are accidentally reused between the frames. This process can be repeated for an arbitrary number of times, allowing acquisition of arbitrarily many images.

Note that the natural mobility of the larvae makes it impossible to eliminate the occlusions; the specimens will occasionally crawl towards and onto one another, which makes automatic segmentation potentially erroneous. This mobility does have one advantageous effect: it can be considered a form of natural data augmentation. Acquiring a short series of images while the larvae crawl around the rearing box allows capturing every instance in several different poses, yielding more variation in larvae shape at no cost. In this study, up to three images in a series were acquired before discarding the larvae and refilling the box.

Since the validation set (used for the validation step during the training of the models) should be independent of the training data, care must be taken not to reuse larva instances between those sets during data synthesis. The recommended precaution is to split the physical larvae into training and validation instances and ensure that division is maintained during data acquisition.

c: Segmentation

In order to facilitate synthesizing images, the individual larvae need to be extracted from the acquired raw images in the process of segmentation. The seemingly straightforward task of segmenting the image of the larvae in the rearing box was complicated by the presence of shadows caused by bidirectional illumination. Thus, instance segmentation for each training image proceeded through several stages. The most distinct separation of instances from the background in the image was noted in the saturation channel of the HSV model and the blue channel of the BGR model, therefore, these two domains were employed for segmentation. For the initial, rough segmentation, global thresholding based on saturation was applied, with the threshold defined experimentally. The determined binary objects were filtered by their area, thus eliminating those smaller than a single insect. Rectangular regions of interest were specified for the remaining detected objects (Fig. 4a). Within each region, saturation-based thresholding was applied once more, but this time, the threshold was adaptively set based on the local histogram. This produced an image with initial binary masks (Fig. 4b), which were further refined in the subsequent step. Adaptive erosion was performed along the edges of the masks, this time using the

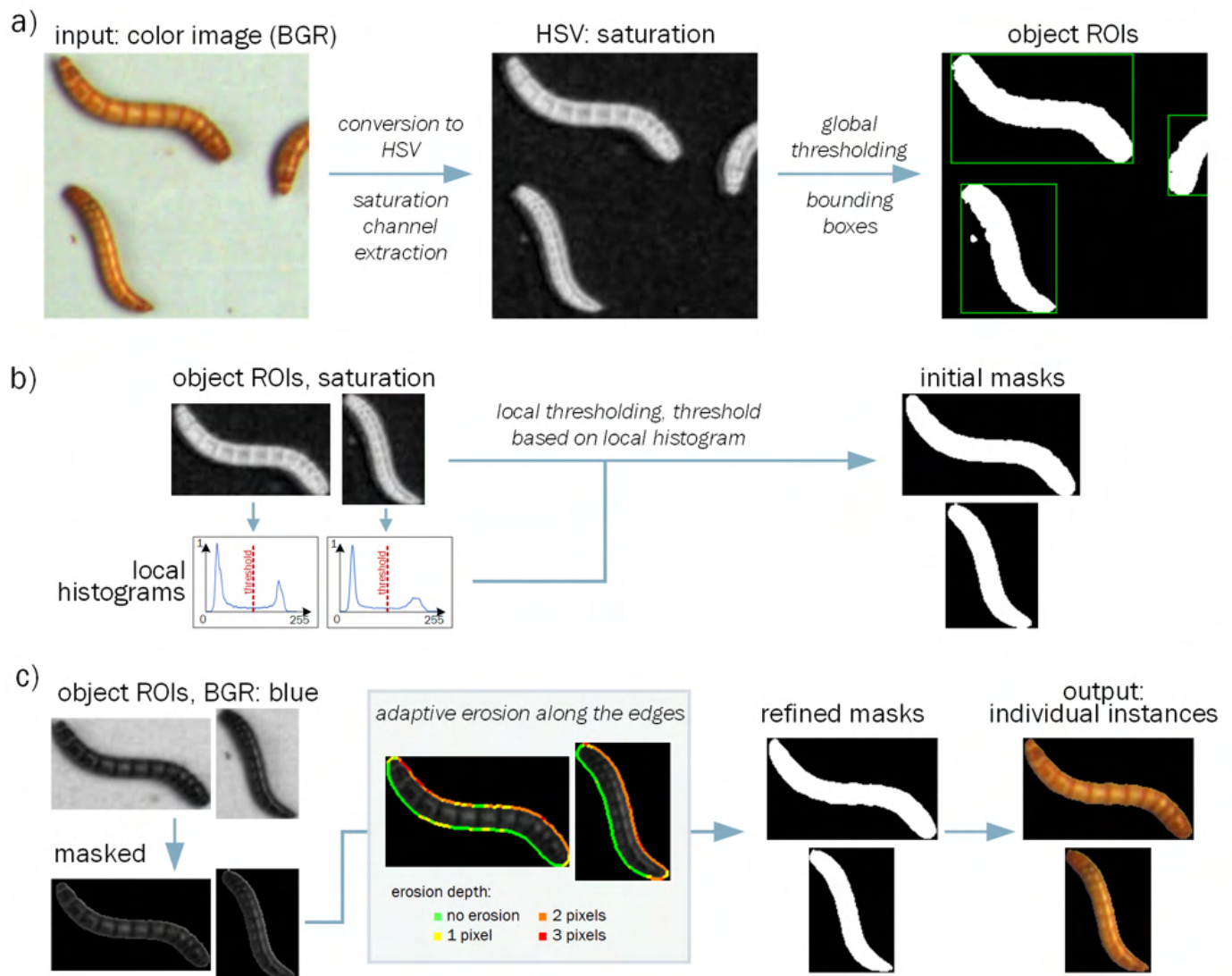


FIGURE 4. Proposed concept of multi-stage segmentation: a) rough segmentation, by thresholding in the saturation channel of the source image converted to HSV model, defining the regions of interest (ROI); b) segmentation in ROIs by thresholding in the saturation channel with the threshold defined based on local histograms, resulting in initial binary masks; c) binary mask refining by performing adaptive erosion along the edges of the initial masks, where the depth of erosion was determined using the blue channel from the source image.

blue channel of the training image. The depth of erosion was determined by the difference between the average value of the kernel covering the nearest neighborhood around the edge point and the average value of the entire object. The final segmentation masks obtained (Fig. 4c) were ultimately subjected to visual inspection and, where necessary, manual correction. For example, the rare cases of overlapping larvae were manually removed from the dataset.

3) Test dataset

a: Concept

Synthetically generated images can be useful to train AIS models, but to validate the complete pipeline, real images with amodal ground truth annotation are required. As previously mentioned, acquisition and manual annotation of real images containing high object density and occlusions

is difficult and expensive. The following sections propose a data acquisition procedure that alleviates this challenge. Again the concept is to reduce the amodal annotation task to modal annotation, making the task of the human annotator significantly easier at the expense of a more time-consuming data acquisition - similar to the approach taken by [28]. The result is a dataset comprising real images, i.e. with no need for synthesis, yet equipped with a complete and correct amodal annotation.

b: Acquisition

Assuming a scene contains no self-occlusions (i.e. instances of objects with complex shapes, whose one region occludes another) or cyclic occlusions (i.e. pairs or tuples of objects that partially occlude each other mutually), it can be decomposed into a layered structure. In this case, instances can be

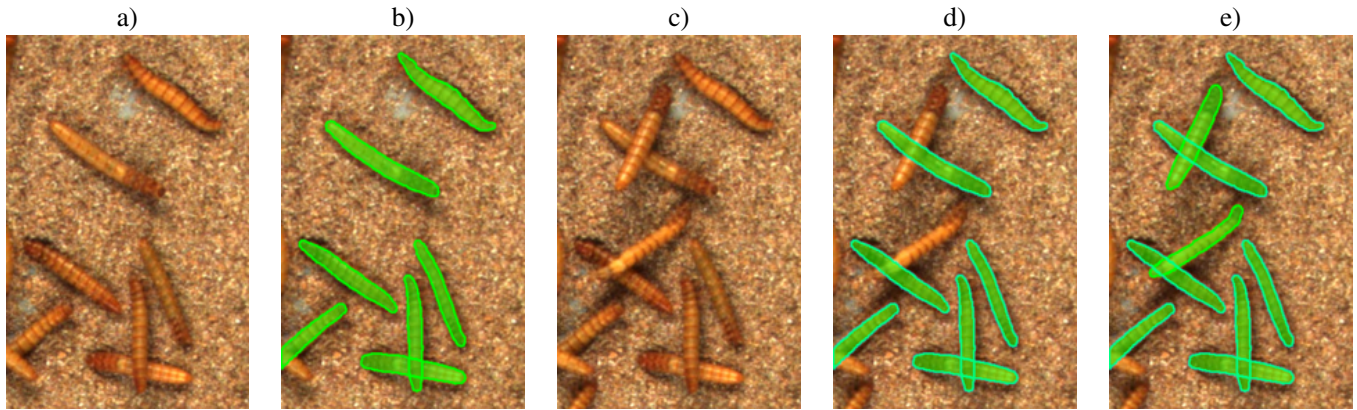


FIGURE 5. Overview of the process of manual annotation of the test set: a) source image of a layer i , b) human-input modal annotations for layer i , c) subsequent frame: layer $i + 1$, note that more objects become occluded, d) modal annotations from frame i become amodal annotations in frame $i + 1$, e) annotator only needs to input the missing modal annotations in frame $i + 1$.

grouped into a sequence of layers by their ordering along the optical axis in such a way that the first layer contains only fully visible instances, and every subsequent layer of index i contains instances that *would be* fully visible if all layers of index j such that $j < i$ were removed. As discussed previously, it is difficult to provide amodal annotation for an image of a heavily occluded scene. But if this image could be decomposed in the aforementioned way, annotation of each layer separately would become a straightforward modal annotation task.

The proposed method of test data acquisition builds on the concept of decomposition into layers. The fundamental assumption is that the acquisition process can be performed in strictly controlled conditions, as is the case with the problem of mealworm larvae monitoring and many other industrial applications. Instead of capturing a single image of the entire scene, the proposed procedure is to capture a sequence of images encoding information about the layer composition. First, acquire the image of the complete, dense scene full of occlusions. Then, physically remove all fully visible instances from the scene, and acquire the image. By repeating those two steps until there are no more occlusions, one obtains a sequence depicting a single scene exploded into individual layers, which are easy to annotate by a human. The result of this annotation is a single test image. The entire procedure needs to be repeated as many times as required to build a test dataset.

Alternatively, this process can be carried out in reverse, by *placing* objects in the scene between acquisitions instead of removing them, building up a progressively more dense scene instead of decomposing it. This approach was taken in the course of this work.

Note that a test set constructed in the proposed manner should be acquired in lighting conditions closely matching that of the deployment environment for the performance metrics measured over this test set to be predictive of the actual performance of the deployed model. Furthermore, the success of the test set construction procedure depends on the assumption that the objects do not move nor change shape in its

course. For inanimate objects, this assumption only constrains the manual precision of the operator, i.e. that they do not accidentally displace underlying objects while manipulating the current foreground layer. However, in the case of mobile, living organisms such as mealworms, this poses a challenge, because the individual specimens exhibit continuous and non-negligible motion. During preparation for this study, larvae designated for the test set were physically frozen prior to data acquisition in order to inhibit all motion.

c: Annotation

Annotating images acquired through the decomposition procedure is substantially easier, as the annotator has access to the entire structure of the scene at all times. Therefore, the annotation uncertainty issue is solved, since every object will be fully visible in at least one image in the sequence, independently of the density of the original scene and the degree of occlusion. To leverage this property and streamline the annotation process, a custom annotation tool is implemented. The annotator begins by marking the boundaries of the fully visible objects, proceeding through the sequence of progressively more occluded images. The tool automatically transfers the completed annotations, so that the annotator only ever marks the foreground objects. Should any of those transferred annotations be incorrect, for example, due to displacement of the objects between subsequent frames, the tool provides a set of simple corrective commands based on affine transformations. At the end of the sequence, all of the objects have received the amodal annotations, despite the annotator always having worked in the modal domain. A visual overview of the process is illustrated in Fig. 5.

C. DATA GENERATION

1) Principle

The objective of the data generation procedure is to create arbitrarily many new images containing various arrangements of larvae, including occlusions, but with complete amodal annotation for every object. This is possible thanks to the data acquisition method, which allows the construction of a

Algorithm 1 Image generation algorithm

Input: database of backgrounds, database of instances, generation parameters image $\leftarrow$ sample_background(database of backgrounds, parameters) instances $\leftarrow \emptyset$ repeat new_instance $\leftarrow$ sample_instance(database of instances, parameters) if placement_correct(new_instance, parameters) then insert new_instance into instances end if until have enough instances for each instance in instances do image $\leftarrow$ draw_instance(image, instance, parameters) end for annotation $\leftarrow$ assemble_labels(instances) return image, annotation	$\triangleright$ see section III-B $\triangleright$ see section III-C6 $\triangleright$ see section III-C2 $\triangleright$ see section III-C3 $\triangleright$ see section III-C4 $\triangleright$ see section III-C5
---	---

database of fully visible instances. The generation of synthetic images consists of several stages, outlined in the general form in Algorithm 1.

First, the function `sample_background` samples the background from a database of background images described in section III-B2a. Next, a set of instances - images of larvae extracted from the background, see section III-B2b - is sampled from the database of instances and randomly placed in the new image space using the function `sample_instances`. The function `placement_correct` verifies whether the object was placed in a way that satisfies the constraints. Instances are then drawn in the image space using the function `draw_instance` built of simple computer graphics tools, resulting in realistic-looking images achieved at low cost, without the need for sophisticated rendering techniques or computationally expensive generative methods. The simplicity of this procedure is such that all operations required to implement it are available in the OpenCV library. Amodal labels for the resulting image are constructed simultaneously by the function `assemble_labels`, resulting in a fully annotated image. The process can be repeated an arbitrary number of times, resulting in a dataset of images of arbitrary size; parameters can also be controlled, allowing the generation of images with a desired object count, density, and occlusion intensity. Individual steps of the algorithm are discussed in greater detail in subsequent sections.

2) Background sampling

The first step of image synthesis is background sampling. The goal is to create images with different background textures, and further augment them to allow the vision model to become more robust. This process begins with a random selection of a source background image. Since the source image is of a high resolution, a crop is taken from this image in a random location and of random size (constrained by parameters). The cropped region is then resized to the desired output image size using bilinear interpolation. Additionally,

the resulting image can undergo further augmentation in HSV color space (saturation and brightness jitter) to further enrich the resulting dataset. Overall, the random resize operation ensures that backgrounds with different texture frequencies will appear in the resulting dataset, and the HSV augmentation prevents the vision model from overfitting to a particular brightness/saturation pattern.

3) Instance sampling

The second, and most important step of the synthesis procedure is object sampling and placement. This is an iterative process, that is, objects are sampled and placed in the image space sequentially. To place a single object, a random larva instance is selected from the database created in section III-B2b. Augmentations are applied in the form of rotation by a random angle ϕ , scaling by random factors s_x and s_y , and horizontal and vertical flips. The instance is then placed in the image space by randomly selecting translation offsets t_x and t_y .

Those offsets can be chosen in one of two modes. In a *fully random* mode, t_x and t_y are chosen freely, with the only restriction being that the transformed instance must be fully contained within the image space. In a *seeded random* mode, first M instances - “seeds” - are placed fully randomly, and locations for all subsequent instances are sampled within a predefined radius R around one randomly selected *seed*. This radius R depends on the approximate diameter of the seed instance (defined for simplicity as the average of its height and width) multiplied by a factor r_{seed} which is a parameter of the method. This ensures that the final image will contain areas of high density as well as sparsely populated areas of exposed background. Overall, the process of instance placement can be thought of as an affine transformation by a matrix A of the following structure:

$$A = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

The last stage of object placement is validation. Maximum allowed occlusion ratio is defined to ensure that no “degenerate” instances - that is, instances completely occluded by a large number of subsequent instances - will be rendered. After placing every instance, a check is performed whether the object currently being placed does not violate the constraints. If that happens to be the case, the object is discarded and the iteration is repeated. Otherwise, the procedure continues until the desired number of instances are successfully placed. Note that this entire step has no visual result: all of the object placements are performed abstractly, which conserves processing time on drawing potentially incorrectly placed instances that would be discarded anyway.

4) Image generation

The final step in the image generation procedure is drawing the final image, i.e. overlay the previously sampled and transformed instances onto the background. The challenge is to generate images that closely resemble the real images, allowing the transfer of a model trained on synthetic data onto the *real* domain, but do so with minimum effort, i.e. without the need for complicated and expensive data acquisition procedures and sophisticated rendering techniques, or computationally expensive generative models. This work proposes a simple but effective procedure involving “drawing” understood as a pixel-wise transfer of the source image (e.g. larva instance) onto the target image (resulting synthetic image) using affine transformations and a blending mask. It is assumed that a list of instances (sub-images of larvae cropped from the background using a mask or contour) together with transformation parameters (described in the previous section) is available. The procedure, illustrated in Fig. 6, consists of the following steps executed consecutively for all instances to be drawn:

- 1) Simulate the instance casting a shadow on the underlying objects and background by drawing a transparent, black region corresponding to the instance location but offset by a random vector.
- 2) Draw the actual instance by applying affine transform A and copying the resulting pixels onto the target image, using the instance mask as a blending mask, i.e. to select the pixels belonging to the object and not its background.
- 3) Reduce the effect of sharp edges of the instance region by applying Gaussian blur to the instance mask before drawing.

Additionally, instances are further augmented during drawing by HSV jittering similar to background augmentation.

Because objects are drawn in the order of sampling, the first instances will appear closer to the background, and thus

be more occluded, than instances sampled last, which appear fully in the foreground. This ensures strictly deterministic z-ordering of the drawn instances, allowing extraction of occlusion information such as occluder-occludee relationships and occlusion ratio.

5) Label assembling

Throughout the entire instance sampling and placement procedure, complete information about each object is retained. This means that regardless of the visibility state of each object (occluded or foreground), its contour is fully known at all times. This contour with all augmentation and placement transformations applied to it constitutes the amodal annotation. Furthermore, due to the total ordering of the instances during sampling, it is trivial to determine the visible and occluded areas of each instance by simply iterating over all subsequent instances and performing bitwise *and* operation on their masks to detect the occlusions. By subtracting all occlusion regions from the original mask, the visible (modal) mask is obtained in addition to the original amodal mask. This is useful, as many AIS models require both the amodal and modal labels for training. The label assembly process is integrated into the image generation procedure and is carried out simultaneously with it, resulting in a COCO-style annotation for each synthetic image, both for the training and validation subsets.

6) Controlling the process

The image generation process described above is a generic procedure that could accommodate many different amodal instance segmentation tasks. Controlling the structure of the resulting images can be achieved by varying the following parameters:

- the output size dimensions H and W (pixels)
- desired number of instances N
- number of instances to be placed in seeded random mode M
- seeded placement radius factor r_{seed}
- maximum allowed occlusion ratio o_{max}

Those parameters can be tuned to steer the generation process towards achieving images that are more or less “challenging” from the amodal instance segmentation point of view. By placing more instances on the image (higher N) one usually obtains a more densely populated scene. However, the synthesizer will ensure not to create larger occlusions than o_{max} , so if more difficult images are desired, both the o_{max} and N need to be raised, as those quantities are related. $o_{max} = 0$ allows no occlusions to occur, 1 would admit occlusions that completely hide instances from view.

Controlling the relation of M to N allows the resulting synthetic images to contain high-density areas with multiple occluding larvae, as well as regions of exposed background. This is crucial for training RCNN-based detection models in particular, as they require a balance between foreground and background pixels. Naively distributing instances uniformly

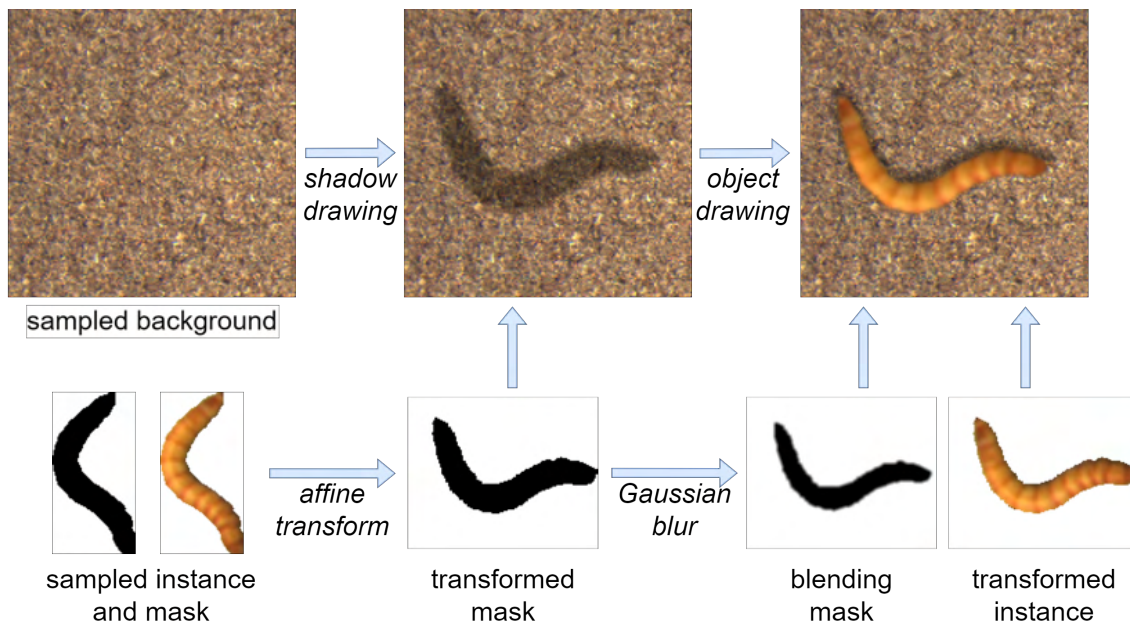


FIGURE 6. Conceptual overview of the image generation procedure. First, a background and a set of instances (a single instance shown for brevity) are sampled. Masks and instance images are transformed using a random affine transform. The transformed mask is then used to draw a shadow in the image space. Next, a blending mask is constructed by blurring the instance mask with a Gaussian kernel. Finally, the instance image is blended with the image via this blending mask. This process is repeated for each sampled instance, producing a final synthetic image.

within the image space while attempting to achieve high-density regions would necessarily cause a significant bias toward object presence, potentially causing false positives.

D. AIS MODELS

1) Mask R-CNN

Mask R-CNN [9] is a common meta-architecture for object detection and instance segmentation. Following the Region-based CNN (R-CNN) scheme, this two-stage system begins with a feature extraction step performed using a backbone model (typically a deep CNN pre-trained on a large image dataset). Feature extraction is followed by a Region Proposal Network (RPN), which generates a fixed number of regions of interest - potential candidate detections - along with the predicted probability of each region containing some object of interest. Those regions are subsequently filtered using a Non-Maximum Suppression algorithm (NMS) to prune the proposal set to only the most likely candidates and thus reduce the computational workload of the second stage of the predictor. Proposals from this reduced set are independently fed to the prediction head, following a RoIAlign operation, to predict class probabilities, precise bounding box coordinates, and segmentation mask.

As a simple and thus easily extensible framework, but nonetheless offering high performance, it has become a foundation for several models designed specifically for amodal instance segmentation tasks. Throughout this work, Mask R-CNN is used as a baseline model in two ways. Firstly, in order to assess the impact of amodal segmentation on the results of mealworm phenotyping, a vanilla Mask R-CNN model will be trained on *modal* labels only. This corresponds to a

typical instance segmentation approach, where occlusions are ignored and only the visible regions of the objects are to be predicted. Secondly, to measure the relative performance of the dedicated AIS models, another instance of Mask R-CNN will be trained on the amodal dataset. This is possible without any modifications to the model because for each instance both the modal and the amodal mask are contained within the same bounding box. Comparing performance achieved by more sophisticated models against this simple reference will help contextualize the results.

2) ORCNN

In [8], the authors proposed the first end-to-end AIS method, setting a strong baseline on existing generic amodal instance segmentation datasets. Their model, Occlusion R-CNN (ORCNN) extends Mask R-CNN with an additional segmentation head dedicated to amodal mask prediction. This allows it to simultaneously predict modal and amodal masks for an object, along with the mask corresponding to the occlusion region, via pixel-wise subtraction of amodal and modal regions. The proposed approach features a special loss function that makes additional use of this occlusion mask for training regularization. ORCNN was successfully trained on datasets *COCO amodal* (COCOA) and *D2S amodal* (D2SA). This model was chosen for comparison due to its simplicity and light weight, traits which could be highly desired in an industrial environment requiring high processing throughputs.

3) Shape Priors

In [13], the authors observed that humans are capable of inferring invisible regions of partially visible regions thanks

to their ability to associate the currently perceived instances to previously seen object classes. In other words, brains encode some form of a shape prior that they use in the process of perception. The authors propose an explicit formulation of the notion of a shape prior in the form of a learned *codebook* containing embeddings representing category-specific shapes. Their model functions as a plug-in to Mask R-CNN, utilizing the same meta-architecture, only extending the prediction head with a dedicated mechanism for amodal segmentation.

In overview, this mechanism consists of three key components. The *coarse mask prediction* module extracts visual features from the proposal regions and predicts a coarse, potentially incorrect segmentation mask. The mask auto-encoder, pre-trained on the amodal ground truth masks, models the shape priors of known object categories. The amodal mask segmentation module merges the coarse mask prediction with the shape prior information, producing the final amodal segmentation. Training of this model is a two-stage procedure. First, the auto-encoder is trained on all amodal segmentation ground truth masks. Then, all masks are encoded using the encoder part, and their collected embeddings are clustered using the k-means algorithm. Cluster centers retrieved this way constitute the codebook. In the second stage of the training, the rest of the model - the coarse mask predictor, the amodal mask predictor, etc. - are trained as usual. At inference, the codebook is used to refine the coarse predictions by passing them through the encoder, and finding the L2-nearest embedding within the codebook, then finally decoding this vector instead of the original embedding.

Several techniques are used on top of that structure to further boost the performance: feature matching, visible attention, reclassification regularization, and others. All are found to improve both the amodal as well as visible mask prediction quality, as demonstrated in a set of ablation tests performed on the D2SA and KINS datasets. The high results on benchmark datasets as well as the explicit formulation of the model, potentially allowing the explanation of its predictions to some degree, were the main reasons for the inclusion of this model in the comparison.

4) AISFormer

In [16], the authors noticed that relationships between visible and invisible regions, or in other words, between occluders and occludees, could be utilized to improve amodal segmentation. They argue that previous Mask-R-CNN-based approaches do not allow modeling of relationships between different objects in the scene, and decide to build their approach, dubbed AISFormer, on the DETR framework instead ([17]). DETR allows capturing even long-distance relationships between various regions, excelling at recognizing which exact object those regions belong to, hinting at the possibility of learning priors that could help in amodal segmentation. The key idea of the AISFormer is thus to simultaneously predict not just two - modal and amodal - but four masks per instance: modal, amodal, invisible, and occluder masks.

AISFormer retains the first stage of the Mask R-CNN, that

is: the backbone, RPN and ROIAlign, but completely replaces its prediction head with a transformer-based module. The mask transformer encoder, operating on image features and positional encoding, models long-range dependencies in the feature space. Following DETR, a mask transformer decoder with learnable query vectors is employed for object detection and modeling the occluder-occludee relationships. An MLP-based invisible mask embedding module models relationships between the visible and amodal masks. Finally, the mask prediction module outputs the four segmentation masks for each instance.

The entire model is trained end-to-end using cross-entropy loss. However, to facilitate the training, it requires special pre-treatment of the dataset: the occluder mask needs to be computed and attached to each instance. AISFormer has matched or exceeded state-of-the-art performance on COCOA, D2SA, and KINS datasets, and the ablation study showed significant performance improvement resulting from the inclusion of occluder mask prediction. The high performance metrics justify the inclusion of this model in the experiments.

E. EVALUATION

1) Detection and segmentation

In the complete pipeline proposed in this work, the amodal instance segmentation model is first trained on synthesized images containing various densities of mealworms and then evaluated on the real test set. Its predictions - a list of detected instances and their amodal masks - are then used as input to the phenotyping algorithm. Therefore, two levels of evaluation are necessary: a vision task evaluation to assess the performance of detection and amodal segmentation, and the phenotyping task evaluation to assess the geometric measurement accuracy. Several models will be trained on the same synthetic dataset and their performance will be compared across the above domains.

Before any evaluation can take place, all trained models must undergo inference parameters calibration, most importantly the value of the NMS threshold in the RPN stage. This is accomplished by evaluating each model on the synthetic validation set several times, over a sequence of predefined threshold values. For the actual performance evaluations (i.e. on the test set), the NMS threshold is set with the value for which the detection performance was the highest. Alternatively, one could optimize the threshold for segmentation performance, but in practice the metrics usually correlate in the function of the threshold, thus a single threshold value maximizes both the detection and the segmentation quality.

Evaluation in the vision domain is performed for object detection and for amodal instance segmentation separately. To measure detection quality, the Average Precision (AP) metric is used, for brevity limited to reporting the AP value at the IoU value of 0.5 (AP50) as well as the mean value (mAP) defined as a mean across thresholds [0.5; 0.95] with a step of 0.05. Segmentation performance evaluation focuses solely on the amodal component, as the modal prediction is of no business value in this application - measurement of only the visible

regions of the larvae yields no information useful from the point of view of industrial process control. The metric used to quantify segmentation performance is also AP, with AP50 and mAP values reported for the test set.

2) Larvae phenotyping

The term “larvae phenotyping” in this article should be understood as determining the length of larvae at the level of individuals and the population as a whole. The key output of larvae phenotyping at the population level is the determination of the histogram of larvae length. For this work, larvae lengths were determined based on binary masks, using the skeletonization method proposed in the article [39]. The intersection coefficient of histograms was used to evaluate the task of phenotyping the larvae population. Suppose that after phenotyping performed for binary masks representing ground truth, a histogram h_{true} was obtained, while for predicted binary masks, a histogram h_{pred} was obtained. Both histograms were characterized by the same bin intervals and were normalized (the sum of all frequencies for each bin equals 1). Let’s denote bins sequentially by the indexes $i = 1, \dots, n$. The histogram intersection coefficient $D(h_{true}, h_{pred})$ is defined as follows:

$$D(h_{true}, h_{pred}) = \sum_{i=1}^n \min(h_{true}^i, h_{pred}^i). \quad (2)$$

IV. EXPERIMENTS

A. PROTOCOL

The main objective of the experiments was to determine whether the models trained on images generated using the proposed method could achieve acceptable performance on the real data. A secondary goal was to rank the models in terms of performance, with particular interest in assessing the impact of amodal completion on phenotyping accuracy. Finally, the results were dissected to identify the factors impacting the phenotyping performance. Details of the experimental procedure follow in the subsequent subsections. Training and test data parameters are summarized in section IV-A1, and model hyperparameters are given in section IV-A2.

1) Data

For the training set, 20 images were acquired, from which 1681 unique instances were automatically extracted. At this stage, they were split into training and validation subsets, containing 1216 and 465 instances respectively. Training and validation datasets were generated in 3 sub-groups of increasing object density and occlusion intensity. Each group contained 500 images: 400 for training, and 100 for validation. Image dimensions were $H = 1400$, $W = 1600$. Generation parameters were established in a series of preliminary experiments, with the objective of replicating the structure of occlusions in the real data. The final parameters are given in detail in Table 1. The resulting dataset contained 1200 training images containing 108,000 instances, and 300 validation images containing 27,000 instances.

TABLE 1. Data generation parameters.

Group	N	M	o_{max}	r_{seed}
“easy”	60	40	0.15	2.0
“medium”	90	45	0.25	1.5
“hard”	120	45	0.45	1.25

To prepare the test set, 5 sequences of images were acquired according to the procedure described in section III-B3, totaling 25 images. Images were cropped to a common region of interest of dimensions $H = 1400$, $W = 1600$ (equal to the training set). Manual annotation was then performed, yielding 2413 instance labels. All of the images were included in the test set. Since the first image in each sequence contains the fewest instances, and the last is the most populated, using all images results in a dataset containing different levels of occlusions. The number of instances per image varied from 48 to 161, with a median of 91.

2) Training hyperparameters

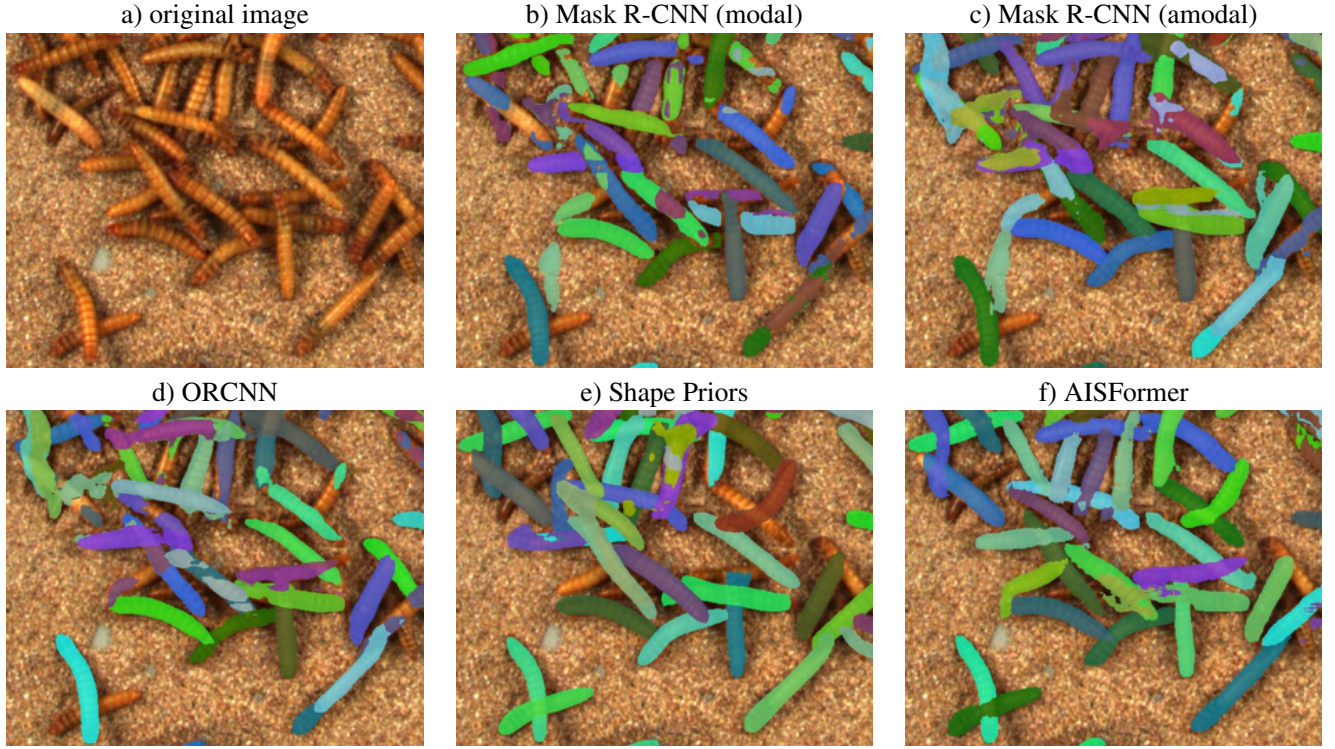
Models trained in this study were taken from their published repositories whenever possible. This included Shape Priors and AISFormer, with implementations based on the Detectron2 framework. The reference implementation of Mask R-CNN from PyTorch’s `torchvision` version 0.13 was used. ORCNN was reimplemented on top of the Mask R-CNN; the validity of the implementation was confirmed by matching performance on the D2SA dataset.

To establish a fair ground for comparisons, all models were trained using the same backbone, a pre-trained ResNet-50. Hyperparameters of the Mask R-CNN meta-architecture, including FPN and RPN configuration (including the anchor generator), were left at their default values according to the PyTorch-based reference implementation. For each model, the strongest version (as described in the respective publications) was used, with all model-specific parameters kept at their optimal values except where noted. ORCNN architecture followed the original publication closely, with four 3x3 convolutions and a ReLU activation in each segmentation head. Shape Priors model was used with all features activated: shape prior refinement, shape prior post-processing, visible attention, feature matching, and reclassification regularization. Codebook clustering was conducted with $k = 128$ clusters, value reduced from the original $k = 1024$ to speed up the process (no loss of performance was observed due to this change). The AISFormer model was used with the default, strongest configuration, i.e. with all four query embeddings.

The choice of learning hyperparameters mostly followed the published reference implementations. Thus, SGD optimizer was used for all models except the AISFormer, which was trained using AdamW. The momentum coefficient was consistent at $\mu = 0.9$. Due to the GPU memory limitations, batch size $B = 1$ was used, while for the prediction heads, batch size was $B_{RoI} = 512$ regions per image. To provide a level ground for comparison, all models were trained for 60,000 iterations with the initial learning rate $\eta = 0.005$.

TABLE 2. Evaluation results of the compared AIS models in terms of detection and segmentation mAP and phenotyping performance in terms of histogram intersection, reported for the test set.

Model	NMS _t	Detection		Segmentation		Histogram intersection
		mAP	AP50	mAP	AP50	
Mask R-CNN (modal)	0.8	0.3823	0.7775	0.2490	0.6451	0.6756
Mask R-CNN (amodal)	0.8	0.3786	0.7844	0.2678	0.7065	0.6745
ORCNN	0.8	0.3688	0.7746	0.3374	0.7237	0.7477
Shape Priors	0.5	0.4072	0.7976	0.3864	0.7942	0.7625
AISFormer	0.9	0.4616	0.8349	0.4093	0.8331	0.7743

**FIGURE 7.** Example predictions on an arbitrarily selected image from the test set (a). Modal Mask R-CNN (b) consistently fails to predict occluded instances, at best predicting the separated halves. Its amodal variant (c) is visibly better, but false positives, false negatives, and segmentation errors can still be observed, in dense regions in particular. Segmentation quality is better for ORCNN (d) but dense regions remain a challenge. Shape Priors (e) greatly improve segmentation quality, and AISFormer (f) adds an even higher detection quality. Note that the z-ordering of instances is not explicitly predicted by any of the models.

achieved after a period of linear warm-up over 1,000 iterations. The learning rate was decayed by a factor $\gamma = 0.1$ after 42,000 and 56,400 iterations. L2 weight decay was used with the factor $\lambda = 0.0001$. Those values were chosen during the preliminary experiments, such that they allowed all models to converge to reasonable values of their loss functions. Ideally, a grid search over the training hyperparameters could be conducted, but this was deemed too computationally expensive as well as beyond the scope of this research. All experiments were conducted on a server equipped with an Intel Xeon Silver 4108 CPU and Nvidia GTX 1080 Ti GPUs.

B. AMODAL INSTANCE SEGMENTATION

After training the models on the training dataset, models were individually calibrated on the validation dataset. This process involved selecting a value for the NMS threshold NMS_t (at the RPN level) that maximized detection mAP. In practice, detection mAP and segmentation mAP correlated, hence the

optimal NMS threshold value usually maximized both measures. To reduce compute workload, threshold values were evaluated in $[0.5; 0.9]$ range, with 0.1 resolution. Preliminary experiments have shown that this resolution was sufficient to determine reasonable values of the threshold since differences between mAP values for thresholds close to optimal were not significantly different. Additionally, to allow testing on the most dense images, the limit of the maximum number of detections per image was raised from 100 to 200. Detection and segmentation results, as measured on the test set, are given in Table 2.

Unsurprisingly, Mask R-CNN trained on the modal labels performs poorly on the test set, which features amodal ground truths only. Simply using amodal labels for Mask R-CNN training boosts segmentation performance substantially. However, while Mask R-CNN is able to complete the invisible regions of the object masks to a limited degree, it is consistently outperformed by all dedicated AIS models.

Segmentation quality improvement is more significant for state-of-the-art models such as Shape Priors and AISFormer. One possible reason for this is that those networks attempt to directly model the concept of a prior - either explicitly, as in the Shape Priors approach, or implicitly via the attention weights of the AISFormer. In contrast, the ORCNN simply offers more network capacity dedicated specifically to the amodal prediction. While this guarantees performance improvement over a naive approach, it proves insufficient to capture more complex relationships between modal and amodal shapes.

Except for the ORCNN, whose detection performance is at best on par with Mask R-CNN, models that perform better at segmentation also tend to be stronger detectors. The built-in ability of the more sophisticated models to directly capture the object structure might be behind their improved detection rate. This is because even for the same set of region proposals, the final prediction (“object” vs “no object”) is made by the model-specific prediction head. In the Shape Priors model, it is the *reclassification module*, while in DETR-inspired AISFormer each prediction head outputs object probability along with the bounding box. Therefore, for more difficult (e.g. occluded) instances, where a simple model would fail to capture the object structure and thus not only fail to segment it correctly but also assign it a low probability, a more sophisticated model might still be able to recover the instance and make the detection.

Fig. 7 contains example predictions for an arbitrarily chosen region of a single image from the test set. Both detection and segmentation improve when more sophisticated models are applied. Even training Mask R-CNN on amodal labels (Fig. 7c) significantly improves the visual quality of the prediction, however, certain errors remain (e.g. false positive in the bottom-left, false negative in the bottom-left and top-right, inconsistent segmentations in the dense regions). Many of those detection errors persist for ORCNN as well (Fig. 7d). Shape Priors (Fig. 7e) and especially AISFormer (Fig. 7f) significantly improve detection performance, particularly in the more densely populated regions. Their segmentation quality is also much higher: most instances are delineated with smooth, simply connected contours, as opposed to multiple “torn” instances produced by Mask R-CNN and its relatively simple extension ORCNN. This proves the efficacy of the proposed training method. The room for improvement remains, however, as exceptionally dense regions still pose a challenge even for the most sophisticated models, and segmentation failures still do appear there.

C. PHENOTYPING PERFORMANCE

The last column of Table 2 summarizes the results of phenotyping task evaluation, measured by the intersection of the resulting histogram with the ground truth histogram, comparing the value of the intersection metric with the segmentation performance of the respective model. As exemplified by the modal variant of Mask R-CNN, relying only on modal perception disallows achieving good phenotyping performance.

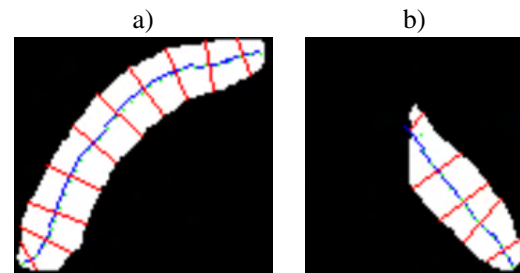


FIGURE 8. Illustration of the behavior of the phenotyping module depending on the quality of segmentation: a) correct segmentation allows correct measurement, b) incomplete segmentation results in a measurement failure.

Partially occluded instances that appear shorter and/or thinner than the fully visible ones directly contribute to measurement error.

Training Mask R-CNN on the amodal labels directly increases the mean average precision of *segmentation*, but this improvement does not help the phenotyping module to make more accurate measurements. Histogram intersection does not change between Mask R-CNN variants in a statistically significant way (Wilcoxon test p -value = 0.426). One possible explanation for this phenomenon might be the nature of segmentation errors that Mask R-CNN makes due to its limitations. Since the phenotyping module measures the length of an instance, the only aspect of amodal completion contributing to its improved performance is the reconstruction of the length of an object (an example of such failure is demonstrated in Fig. 8). Mask R-CNN trained on amodal labels might become more robust to slight occlusions impacting the breadth of a larva, but gain no significant ability to reconstruct its length. This is because Mask R-CNN lacks any mechanism to expand the region of interest bounding boxes, so any segmentations made by this model must be confined to the size of a proposal box, which is generated modally.

Dedicated amodal instance segmentation models all feature such mechanisms, so their ability to reconstruct the length of an occluded instance is much stronger. This allows their performance to scale, leading to a correlation between higher segmentation quality and better phenotyping performance, especially evident in the superior histogram intersection values for the Shape Priors model and AISFormer. All of the dedicated methods achieve significantly greater histogram intersection than either variant of Mask R-CNN (Wilcoxon test p -value < 0.001). Differences between them however are not always significant - only AISFormer consistently outperforms ORCNN (Wilcoxon test p -value = 0.022). For AISFormer versus Shape Priors and Shape Priors versus ORCNN comparisons, differences are not statistically significant (Wilcoxon test p -values = 0.252 and 0.165 respectively).

One can examine the structure of measurement errors by analyzing Fig. 9, which displays histograms computed from segmentation results of all models, overlaying them onto ground truth histograms of the entire test set. In the ideal case, the histograms would perfectly overlap, i.e. every object

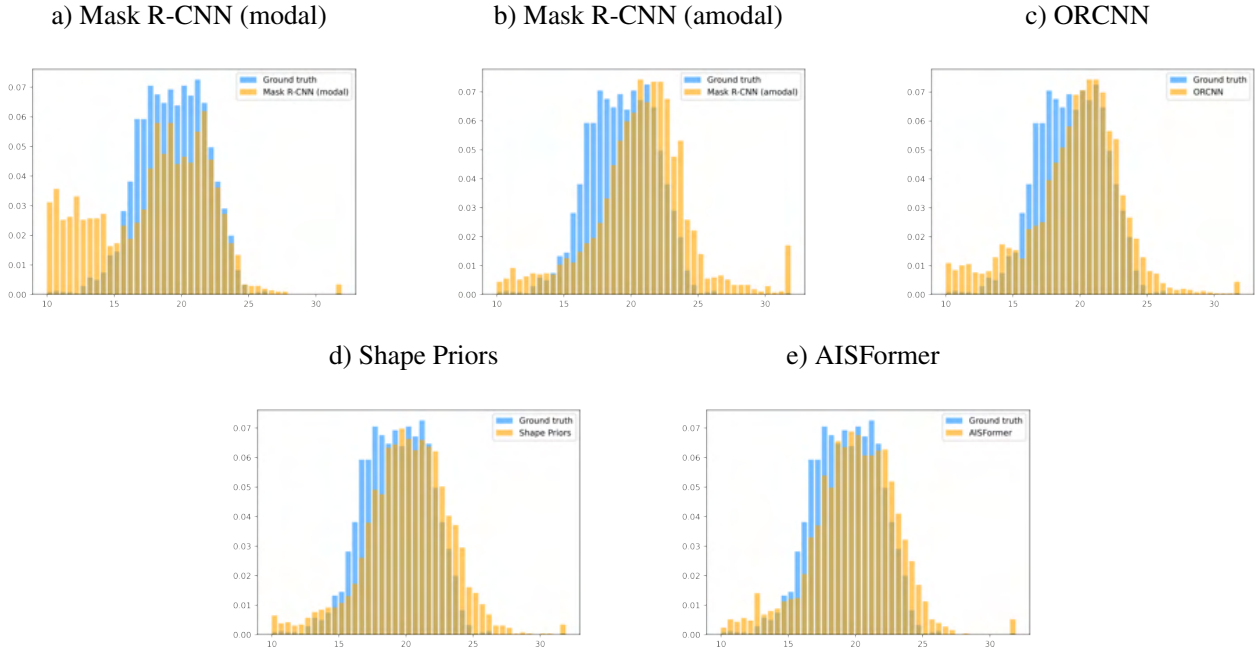


FIGURE 9. Histograms of larvae lengths measured from predictions of different AIS backbones (yellow), overlaid on the test set ground truth histogram (blue). Note an excessive number of short instances in a), and larger histogram intersection of d) and e).

would be assigned a length that places it within the same bin in the histogram (a small error is thus allowed without loss of reported performance). Generally, two types of errors can be defined: a measurement that places an object within a different bin, and a measurement that places the object outside of an expected range of the histogram. That latter mode can be divided into two cases: when the object measures shorter than any expected object, and when it measures longer - those two errors constitute tails in the distributions shown in the figure. Mask R-CNN trained on modal labels (Fig. 9a) predicts a large number of small instances (left tail of the distribution), which is consistent with the theoretical lack of amodal completion ability in the length dimension. Simply training this model on amodal labels significantly reduces the number of errors of this kind. However, the amodal variant of Mask R-CNN assigns too long masks for a large number of instances (right tail of the distribution). This might be due to the inability to distinguish individual instances in high-density regions, and erroneously merging them together. Indeed, it is a frequent mistake (even for human annotators) to attribute a fragment of a large cluster of larvae to a wrong instance, causing its label to appear longer. Errors of this kind are reduced in dedicated AIS architectures, starting with ORCNN. However, some size underestimation errors still appear in ORCNN. The error structure of the more sophisticated models, Shape Priors and AISFormer (Fig. 9 d and e), illustrates the reduction of the number of errors of both kinds.

D. IMPACT OF OBJECT DENSITY

Correlations between detection performance and segmentation quality found in section IV-B, as well as visual assess-

ment of the segmentation masks produced by all models hint that high object density poses an issue. Many detection - and, therefore, instance segmentation - algorithms suffer from precision and recall trade-off, as ensuring a low rate of false negatives (*misses*) results in a growing number of multiple predictions (false positives). This is especially true for R-CNN-based architectures, as they directly relate rectangular regions to instances. Those models feature an explicit instance filtering step - non-maximum suppression (NMS) - in which predictions with a significant bounding box overlap are removed from the result. In the amodal instance segmentation setting, precision-recall balance is even more of a challenge, as the presence of inter-object occlusions necessarily results in greater bounding box overlaps than in the modal case. Therefore, it is of particular interest to quantify the impact of object density - and, indirectly, occlusions - on the overall performance.

To measure the effect density has on the results, the term *density* needs to be defined first. Simply relating the number of instances to the overall area of the image is insufficient, as regions with no instances do not actually contribute to the “difficulty” of an image; in this sense, an image containing a small cluster of tightly packed and heavily overlapping instances would have equal average density to an image with the same number of evenly distributed instances. To avoid this effect, *average density* is defined as the ratio of the total instance count in the image to the total area of the union of their bounding boxes (formula 3, below). For convenience, this quantity will be normalized by the factor of 10^4 , which can be interpreted as the number of instances per 100x100

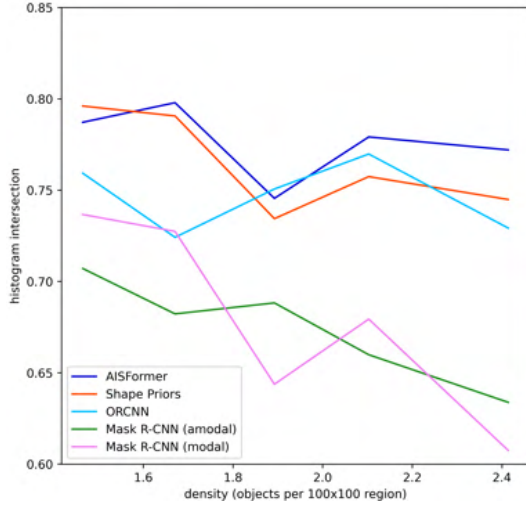


FIGURE 10. Comparison of phenotyping performance as a function of object density, depending on the amodal instance segmentation backbone. Mask-R-CNN-based results visibly degrade with increasing density, while the performance of state-of-the-art models such as Shape Priors and AISFormer remains similar across the examined density range.

pixel region of the image.

$$\rho_i = \frac{n_i}{\|\bigcup_{j=1}^{n_i} B_{i,j}\|} \cdot 10^{-4}. \quad (3)$$

Here n_i is the number of instances in the i -th image, $B_{i,j}$ is the bounding box of the j -th instance in the i -th image, and $\cup$ denotes the union operation.

Examining the impact of average density on phenotyping results is done in two steps. First, the test set is grouped into bins containing images of similar density. Consequently, the phenotyping module is evaluated on images from individual bins and the results are averaged per bin. Results, for different amodal instance segmentation backbones, are plotted in Fig. 10.

All AIS models perform similarly in the low-density regime. Some differences can be observed, as even the low-density images still contain some amount of instances with overlapping bounding boxes. When density - and thus detection difficulty - increases, the performance of the Mask R-CNN models rapidly degrades, regardless of the training mode. There is no apparent difference in the rate of this degradation that could be construed as an argument in favor of either. More sophisticated models such as Shape Priors or AISFormer are superior to Mask R-CNN, but their main strength manifests itself in a much flatter response to growing image density. With Mask R-CNN as a segmentation backbone, phenotyping performance drops by up to 17% when average density increases from 1.4 to 2.4 objects per 100x100 region. For state-of-the-art models, no similar decrease in performance can be observed.

E. LIMITATIONS AND EXTENSIBILITY

The proposed pipeline for generating synthetic data for the training of the amodal instance segmentation models relies

on simple geometric transformations and does not result in photo-realistic images. This causes some visual discrepancy between the synthetic images and the real images acquired using hardware. While this did not pose significant issues in the application scenario of mealworm growth monitoring as demonstrated in this paper, it is possible that reducing this gap might lead to improvements in the segmentation performance. If this was needed, further research might concern implementing more sophisticated rendering methods or including generative models, for example, style transfer into the real images domain.

Although the proposed method is evaluated in the context of mealworm growth monitoring, the principle behind it is generic, and as such it could be adapted to other problem domains. The only limiting assumption is that of a flat, two-dimensional scene structure, as the data generation procedure does not simulate the perspective distortion. Additionally, transferring the proposed method to another problem domain might necessitate the reformulation of the automatic segmentation step, as the procedure described in section III-B2c is specific to the problem of mealworm segmentation. Alternatively, the recent advances in zero-shot segmentation (such as the Segment Anything Model) could be applied to execute this stage of data preparation (those methods were not yet available during the preparation of the training data for this project).

V. CONCLUSION

In this work, the task of phenotyping *T. molitor* larvae was framed as an amodal instance segmentation problem. To overcome the common challenge of obtaining properly annotated data for training AIS models, a novel method for the generation of training data was proposed. Instead of relying on time-consuming and error-prone manual annotation of amodal data, images were acquired in a fully modal setting which removed the need for manual annotation. Simple computer graphics tools were then used to generate images containing occlusions, with complete amodal annotation constructed simultaneously. Additionally, a dedicated method for acquiring real images for the independent test set was introduced, greatly simplifying the annotation procedure and enabling proper testing of AIS models trained on synthetic data. Finally, several generic AIS methods were trained on data generated using the proposed pipeline, and comprehensively evaluated in terms of amodal segmentation performance and accuracy of geometric measurement of the detected larvae.

The key finding was that training the vision models on synthetic data generated with the proposed method allows good segmentation results on the real test data. Furthermore, all amodal models were found to achieve better segmentation performance than a baseline Mask R-CNN trained on modal labels only. However, improvement in the vision domain did not immediately translate to the geometric measurement domain - only the dedicated AIS models enabled substantially better measurement accuracy. This confirmed that the amodal approach has the potential for significant

improvements in mealworm phenotyping accuracy and that the proposed methodology for data preparation is the key to unlocking this potential.

Additionally, correlations were observed between segmentation and detection performance. In general, models that were better at amodal segmentation tended to be stronger object detectors. While dissecting this phenomenon, a relationship between object density and measurement performance was found. Weaker segmentation models performed poorly when density increased, while the strongest models (Shape Priors, AISFormer) offered a more consistent performance curve, showing greater robustness in high-density situations, where the performance gap was the largest.

The demonstrated approach could be applied in an industrial system for remote monitoring of mealworm rearing boxes, allowing more accurate larvae dimensioning. However, since object occlusions arise as a challenge in other engineering applications of computer vision, especially in agriculture and manufacturing, the proposed method could find broader applications.

ACKNOWLEDGEMENTS

We wish to thank the company Tenebria of Lubawa, Poland, for providing the source material - live specimens of *Tenebrio molitor* larvae - for experiments conducted in the course of this work.

REFERENCES

- [1] S. Mancini, G. Sogari, S. Espinosa Diaz, D. Menozzi, G. Paci, and R. Moruzzo, "Exploring the future of edible insects in europe," *Foods*, vol. 11, no. 3, p. 455, 2022.
- [2] P. Majewski, P. Zapotoczny, P. Lampa, R. Burduk, and J. Reiner, "Multipurpose monitoring system for edible insect breeding based on machine learning," *Scientific Reports*, vol. 12, no. 1, p. 7892, 2022.
- [3] W. Gerbino and D. Salmaso, "The effect of amodal completion on visual matching," *Acta Psychologica*, vol. 65, no. 1, pp. 25–46, 1987. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/000169188790045X>
- [4] K. Li and J. Malik, "Amodal instance segmentation," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 677–693.
- [5] Y. Zhu, Y. Tian, D. Metaxas, and P. Dollár, "Semantic amodal segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.
- [7] K. Ehsani, R. Mottaghi, and A. Farhadi, "Segan: Segmenting and generating the invisible," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [8] P. Follmann, R. König, P. Härtinger, M. Klostermann, and T. Böttger, "Learning to see the invisible: End-to-end trainable amodal instance segmentation," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 1328–1336.
- [9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.
- [10] X. Zeng, X. Liu, and J. Yin, "Amodal segmentation just like doing a jigsaw," *Applied Sciences*, vol. 12, no. 8, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/8/4061>
- [11] L. Qi, L. Jiang, S. Liu, X. Shen, and J. Jia, "Amodal instance segmentation with kins dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [12] Z. Zhang, A. Chen, L. Xie, J. Yu, and S. Gao, "Learning semantics-aware distance map with semantics layering network for amodal instance segmentation," in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2124–2132. [Online]. Available: <https://doi.org/10.1145/3343031.3350911>
- [13] Y. Xiao, Y. Xu, Z. Zhong, W. Luo, J. Li, and S. Gao, "Amodal segmentation based on visible region segmentation and shape prior," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, pp. 2995–3003, May 2021. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/16407>
- [14] Y. Sun, A. Kortylewski, and A. Yuille, "Amodal segmentation through out-of-task and out-of-distribution generalization with a bayesian model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 1215–1224.
- [15] L. Ke, Y.-W. Tai, and C.-K. Tang, "Deep occlusion-aware instance segmentation with overlapping bilayers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 4019–4028.
- [16] M. Tran, K. Vo, K. Yamazaki, A. Fernandes, M. Kidd, and N. Le, "Aisformer: Amodal instance segmentation with transformer," *arXiv preprint arXiv:2210.06323*, 2022.
- [17] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.
- [18] H. Lamdouar, W. Xie, and A. Zisserman, "Segmenting invisible moving objects," 2021.
- [19] J. Yao, Y. Hong, C. Wang, T. Xiao, T. He, F. Locatello, D. Wipf, Y. Fu, and Z. Zhang, "Self-supervised amodal video object segmentation," vol. 35, 2022, Conference paper. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85163156150&partnerID=40&md5=30212734e74540923b57133d1a5eb6b5>
- [20] J. Xie, W. Xie, and A. Zisserman, "Segmenting moving objects via an object-centric layered representation," vol. 35, 2022, Conference paper. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85153792009&partnerID=40&md5=427e2252cd06a2d7fe6d423f1a21cd9a>
- [21] R. Mohan and A. Valada, "Amodal panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 21 023–21 032.
- [22] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and Y. Akbari, "Image inpainting: A review," *Neural Processing Letters*, vol. 51, pp. 2007–2028, 2020.
- [23] A. Kar, S. Tulsiani, J. Carreira, and J. Malik, "Amodal completion and size constancy in natural scenes," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [24] X. Wu, Y. Meng, J. Zhang, J. Wei, and X. Zhai, "Amodal segmentation of cane sugar crystal via deep neural networks," *Journal of Food Engineering*, vol. 348, p. 111435, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S026087742300033X>
- [25] O. L. F. de Carvalho, O. A. de Carvalho Júnior, A. O. de Albuquerque, A. S. Luiz, N. C. Santana, and D. L. Borges, "Beyond the visible pixels using semantic amodal segmentation in remote sensing images," in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 310–313.
- [26] N. Aditama and R. Munir, "Indonesian street food calorie estimation using mask r-cnn and multiple linear regression," in *2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T)*, 2022, pp. 1–6.
- [27] Y. Hu, J. Wang, X. Wang, and T. Guan, "Occlusion-aware particle size distribution detection of gravel material based on the improved bilayer convolutional network," *Construction and Building Materials*, vol. 404, p. 133263, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S095006182302980X>
- [28] P. M. Blok, E. J. van Henten, F. K. van Evert, and G. Kootstra, "Image-based size estimation of broccoli heads under varying degrees of occlusion," *Biosystems Engineering*, vol. 208, pp. 213–233, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1537511021001203>
- [29] J. Gené-Mola, M. Ferrer-Ferrer, E. Gregorio, P. M. Blok, J. Hemming, J.-R. Morros, J. R. Rosell-Polo, V. Vilaplana, and J. Ruiz-Hidalgo, "Looking behind occlusions: A study on amodal segmentation for robust on-tree apple fruit size estimation," *Computers and Electronics*

- in *Agriculture*, vol. 209, p. 107854, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169923002429>
- [30] J. Monica, W.-L. Chao, and M. Campbell, "Sequential joint shape and pose estimation of vehicles with application to automatic amodal segmentation labeling," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2678–2685.
- [31] W. Agnew, C. Xie, A. Walsman, O. Murad, Y. Wang, P. Domingos), and S. Srinivasa, "Amodal 3d reconstruction for robotic manipulation via stability and connectivity," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 1498–1508. [Online]. Available: <https://proceedings.mlr.press/v155/agnew21a.html>
- [32] Z. Xu, Z. He, J. Wu, and S. Song, "Learning 3d dynamic scene representations for robot manipulation," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 126–142. [Online]. Available: <https://proceedings.mlr.press/v155/xu21b.html>
- [33] S. Back, J. Lee, T. Kim, S. Noh, R. Kang, S. Bak, and K. Lee, "Unseen object amodal instance segmentation via hierarchical occlusion modeling," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 5085–5092.
- [34] S. Kim, S.-J. Hong, J. Ryu, E. Kim, C.-H. Lee, and G. Kim, "Application of amodal segmentation on cucumber segmentation and occlusion recovery," *Computers and Electronics in Agriculture*, vol. 210, p. 107847, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169923002351>
- [35] Q. Zhou, S. Wang, Y. Wang, Z. Huang, and X. Wang, "Human de-occlusion: Invisible perception and recovery for humans," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3690–3700.
- [36] E. Huang, Z. He, A. Mao, M. Camila Ceballos, T. D. Parsons, and K. Liu, "A semi-supervised generative adversarial network for amodal instance segmentation of piglets in farrowing pens," *Computers and Electronics in Agriculture*, vol. 209, p. 107839, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169923002272>
- [37] Y. Hu, J. Wang, X. Wang, J. Yu, and J. Zhang, "Efficient virtual-to-real dataset synthesis for amodal instance segmentation of occlusion-aware rockfill material gradation detection," *Expert Systems with Applications*, vol. 238, p. 122046, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417423025484>
- [38] A. Baur, D. Koch, B. Gatterner, and A. Delgado, "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*, vol. 8, no. 8, pp. 913–920, 2022.
- [39] P. Majewski, M. Mrzygłód, P. Lampa, R. Burduk, and J. Reiner, "Monitoring the growth of insect larvae using a regression convolutional neural network and knowledge transfer," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107358, 2024.
- [40] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE J. Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.



includes object detection and instance segmentation, especially amodal instance segmentation.

PRZEMYSŁAW DOLATA received the M.Sc. degree in Mechatronics from Wrocław University of Science and Technology, Wrocław, Poland, in 2016. He worked at the Optomechatronics Group, Faculty of Mechanical Engineering, until 2021, researching machine learning approaches for computer vision problems in industry and agriculture.

In 2021 he moved to WUST's Department of Artificial Intelligence in pursuit of a PhD in machine learning and computer vision. His research inter-



and transportation sectors. He is the main author of publications in journals indexed in Philadelphia list and has presented at well-recognized international conferences, including CVPR. Since 2024, he has also been working as a Senior Data Scientist in the commercial sector.

PAWEŁ MAJEWSKI graduated with an Engineering degree in Mechatronics in 2019 and a Master's degree in Big Data Analysis in 2020 from Wrocław University of Science and Technology, Poland. In 2025 he was awarded a PhD in machine learning methods in precision agriculture and insect biosystems engineering. His research interests primarily lie in computer vision. He has served as a machine learning expert and research leader in five machine vision R&D projects across the agriculture, energy,



From 2015 to 2019, he developed an automated barley quality control system. Between 2017 and 2021, he worked on plant imaging to assess nutrient deficiencies. In 2021–2023, he worked on a vision system to monitor the farming of edible insects (mealworms).

His research interests include optomechatronics, machine vision, image processing, and food quality control.

PIOTR LAMPA graduated for Master of Science in Engineering degree in Mechatronics at the Faculty of Mechanical Engineering, Wrocław University of Science and Technology, Poland, in 2014. Since then, as research and teaching assistant, he focused on developing machine vision systems for quality control, especially of natural products and food.



top ML conferences including NeurIPS, ICLR, and ICML. His research interests include deep learning, especially generative models, and representation learning.

MACIEJ ZIĘBA received the master's degree in computer science from the Blekinge Institute of Technology, Sweden, and the master's degree in economics and the PhD degree in computer science from the Wrocław University of Science and Technology. He is currently an AI researcher with Tooploox and an associate professor with the Wrocław University of Science and Technology. He was the co-author of over 90 research papers published in significant journals and presented on the

JACEK REINER is an Associate Professor and the Leader of Optomechatronics Group at Faculty of Mechanical Engineering, Wrocław University of Science and Technology, Poland. His research expertise and interests cover Machine Vision with Machine Learning for Processes Control and Natural Products Inspection, as well as Industrial Laser Technologies for Manufacturing.



Direct detection of elongated objects geometry via a centerline-based representation

Przemysław Dolata¹[0000–0002–0629–8130] and
Maciej Zięba^{1,2}[0000–0003–4217–7712]

¹ Department of Artificial Intelligence, Wrocław University of Science and Technology, 50-372 Wrocław, Poland
<https://ai.pwr.edu.pl>

² Tooploox, 53-601 Wrocław, Poland
<https://tooploox.com>

Abstract. Object detection aims to locate objects in images by classifying them and assigning them bounding boxes. However, this representation is insufficient when details of the object geometry are of interest. In such cases, more computationally intense methods such as instance segmentation are necessary. Furthermore, bounding boxes do not distinguish heavily occluding objects.

In this work, we propose a more geometrically descriptive representation, a *centerline*. We train an object detection model to predict centerlines instead of bounding boxes. This allows us to obtain some geometric properties of the objects directly during detection, without the need for segmentation. We also find the proposed approach to improve detection performance over the bounding-box-based baseline.

Keywords: Object detection, Instance segmentation, Amodal segmentation

1 Introduction

Many applications of computer vision require some form of measurement or estimation of the geometry of the objects. Object detection is the common frame-

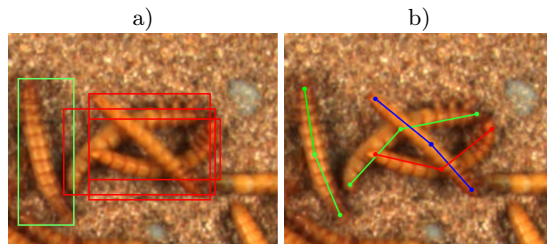


Fig. 1. In heavily occluded scenes, bounding boxes can be difficult to distinguish (a). A better way to represent the object geometry might alleviate this problem while providing more information about the detected objects (b).

work to implement those applications, but detection - defined as rectangular bounding box regression - provides limited information about each instance. In order to perform any detailed reasoning, further processing of the detection results is necessary.

In cases where geometric measurement or shape analysis is of interest, raw bounding boxes are insufficient. Instance segmentation methods are employed, allowing a precise segmentation mask or a contour of the object to be determined. Usually, this is done downstream of detection: first, the bounding boxes are detected, then segmentation is performed within those boxes [1, 4]. However dense prediction tasks, such as segmentation, often require large models that are slow both at training and inference time.

Furthermore, in certain application areas, such as agriculture, detection itself is made exceptionally more challenging due to the unconstrained scene structure. For example, objects might appear in very dense groups, occluding one another significantly. Amodal instance segmentation (AIS) - the task to predict *all* pixels belonging to an instance, including the occluded regions - finds use in those contexts.

However, most AIS models are built on top of common object detection models, which tend to struggle under heavy occlusions. The reason for that is the failure of the fundamental assumption of object detection as a task: that each bounding box uniquely corresponds to one object. While this is true in general-purpose applications, in some specific use cases multiple objects can occupy very similar spatial locations (Figure 1a), making detection via bounding boxes an ill-posed task regardless of the model used. One example of such an industrial problem is mealworm monitoring, where the objects are of elongated shape, possibly curved, and frequently occlude one another [10].

We propose an approach to object detection built on a more expressive geometric primitive than a bounding box: a *centerline*, which can be thought of as a polygonal chain between 3 points. This structure represents elongated objects such as mealworms more accurately than a bounding box. At the same time, it has all the qualities that make detection possible: it locates an instance in space, describes its size, and can be learned using a regression-type loss. Most importantly, it offers a unique object to representation mapping even in heavily occluded scenarios (Figure 1b).

In this paper we reformulate a state-of-the-art detection architecture, DETR, to predict centerlines instead of bounding boxes. We evaluate its performance on the task of mealworm detection, comparing it against the bounding-box-based baseline. We demonstrate the ability of the proposed approach to directly predict a rich geometric representation, as well as the ability to estimate certain geometric properties of the detected objects directly from this representation, without the need for segmentation - a feat that would not be possible with a bounding-box-based detector.

2 Related works

2.1 General object detection

In general, object detection is understood as a task of predicting an unspecified number of rectangular bounding boxes with class labels, given an input image. The most common approaches to this task can be divided into single-stage and multi-stage meta-architectures.

One of the leading examples of a multi-stage detector is Faster R-CNN [14]. This two-stage architecture first performs a coarse search in the image feature space, returning a (possibly large) set of *region proposals*, which are themselves rectangular bounding boxes, with an “objectness” probability score attached. Afterward, the second stage performs the precise bounding box regression and classification by processing a region cropped from the feature tensor according to each proposal. Models based on the R-CNN scheme often offer superior detection performance, and are relatively easy to extend to different tasks such as instance segmentation [4].

On the other hand, single-stage detectors attempt to perform the bounding box prediction without intermediate steps. YOLO, and a large number of its extensions, divides the input image into a predefined grid, making predictions for each region individually [13]. By simplifying the coarse search (region proposals in R-CNN descendants) this way, great speed improvements are achieved. Typically, those methods do not achieve as high detection performance as multi-stage models but are capable of performing detection in real time.

In contrast to the aforementioned approaches, DETR formulates object detection directly as a set prediction task and employs a transformer-based architecture to perform it in a single pass [1]. In this framework, features extracted from the image using a backbone network are interpreted as input sequence tokens. Those tokens are encoded using several layers of multi-head self-attention. The set of outputs is produced by the decoder, consisting of multi-head self-attention and decoder-encoder cross-attention, which receives those transformed tokens and a matrix of learned queries.

The loss function in DETR treats the matching of predictions to ground truths as a bipartite matching problem and solves it using the Hungarian algorithm. This approach not only allows supervision of class prediction and bounding box coordinate regression but also directly addresses the false positives problem common in R-CNN-based architectures. Since only a single prediction can be matched to each ground truth instance, extraneous predictions receive a penalty, forcing the model to output a special **no-object** prediction instead. Note that this would not be possible in a typical two-stage model like Faster R-CNN, since its prediction head makes predictions for each proposal in isolation; DETR’s self-attention however allows queries to exchange information, making it possible for one query to focus on an instance while another outputs **no-object** instead of duplicating the prediction.

2.2 Predicting object geometry

Typically, when specific information on object shape or geometry is required, instance segmentation methods are employed. Mask R-CNN extends Faster R-CNN with a mask prediction head, performing detection first, and then predicting dense segmentation for each detected instance [4]. Additionally, it offers a variant for extracting ordered keypoints, following the same detect-and-predict approach. Directly predicting a keypoint-based representation of the object geometry can be more convenient than predicting a binary segmentation mask and attempting to retrieve geometry from there.

In some cases, the typical horizontally aligned rectangular bounding boxes are not sufficiently rich in information for a given detection task. For instance, in bio-analysis, cell detection and tracking can benefit from a more suitable representation geometry. By parameterizing objects as oriented ellipses instead of rigid bounding boxes, state-of-the-art results can be achieved with much less data annotation effort [6].

Extraction of a skeleton-based representation is useful in tasks such as pose estimation or action recognition, also in the 3D domain. Point2Skeleton is an unsupervised method of learning such representations from point cloud data [8]. In this approach it is assumed that an input point cloud contains exactly one object of interest, and so the skeleton generation does not serve detection, but helps enrich the representation of a (presumably already detected) object.

PolyTransform is another interesting approach to instance segmentation, turning a dense mask prediction into a polygonal shape prediction problem [7]. In this framework, some form of object detection and coarse segmentation is employed, followed by a network that fits a polygon to each object’s boundary. This work focuses on improving segmentation quality and uses a backbone network for object detection and generation of “polygon initializations”, which the subsequent stages will refine into the final geometry. Impressive results are reported, but failures are noted - in particular under complex occlusions, where the underlying detector fails, and the resulting initialization is of poor quality.

Obtaining structural representations of detected objects would be useful in many contexts where object geometry is relevant. In those cases, the usual approach is to perform detection first, and then either segmentation, or some form of geometry prediction. That means the performance of such models depends on the quality of the upstream object detector. Predicting some form of object geometry representation *directly*, i.e. at the detector stage, could be useful in those applications due to eliminating the need for segmentation altogether. Furthermore, a model of this kind could offer stronger detection performance, especially in cases of particularly heavy occlusions, where bounding boxes can potentially contain multiple objects. To the best of our knowledge, no such approach has been published so far.

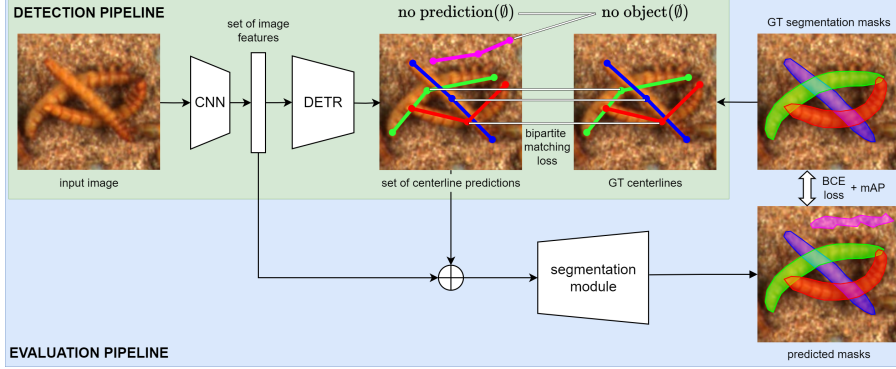


Fig. 2. Overview of the proposed method. Our detection pipeline follows the DETR scheme, but instead of bounding boxes, we predict centerline coordinates directly. We redefine the bipartite matching loss to compute optimal matching between predicted centerlines and ground truth labels generated from dense segmentation masks. Due to a lack of standard evaluation metrics for a centerline-based representation, we evaluate our detection performance in terms of a downstream task, i.e. via segmentation.

3 Method

3.1 Overview

We propose a method of detecting objects directly via a rich, structural representation we call a *centerline*. We assume DETR as a base meta-architecture, replacing the final bounding box prediction layer with a new centerline prediction layer - see Figure 2, green inset. For training this model, we follow the same paradigm of bipartite matching via the Hungarian algorithm, but we redefine the matching loss to compute the similarity between centerlines instead of bounding boxes. We obtain ground truths for training the model by extracting centerlines from segmentation labels.

Ideally, detection performance of the proposed model should be compared to a baseline reference following the bounding box paradigm. However, mAP metrics for bounding boxes inherently depend on the concept of IoU similarity, which is not trivial to define for centerlines. Therefore, we choose to avoid the ill-posed comparison, and evaluate our method on a downstream task instead. We train an instance segmentation model on top of our detector, conditioning the segmentations on centerlines, and compare segmentation mAP to the reference bounding box instance segmentation model. Figure 2 illustrates the scheme.

3.2 Centerline representation

Commonly, object detection is defined as a regression-type task to predict bounding box coordinates (x, y, h, w) or, equivalently, (x_c, y_c, h, w) - depending on whether the box is anchored by its top-left corner or the center.

In our proposed method, we take note of the elongated shape of the objects being detected, and introduce an inductive bias based on that property. We define the regression output as a *centerline*, that is a tuple of three points: a unique center point c , and two endpoints e_1 and e_2 . This yields a 6-component representation $(x_c, y_c, x_{e_1}, y_{e_1}, x_{e_2}, y_{e_2})$. It is important that the endpoints e_1, e_2 should be treated as permutation invariant, that means: tuples (c, e_1, e_2) and (c, e_2, e_1) refer to the exact same instance.

It is possible to retrieve bounding box coordinates from a centerline, but not vice versa. Indeed,

$$\begin{aligned} x_{\text{box}} &= \min(x_c, x_{e_1}, x_{e_2}) \\ w_{\text{box}} &= \max(x_c, x_{e_1}, x_{e_2}) - x_{\text{box}} \end{aligned} \quad (1)$$

where x_{box} is the position of the bounding box in the X axis and w_{box} is its width. Analogous procedure can be done to obtain Y coordinates. Note that endpoint coordinates need not both be extreme points of a centerline in cases of heavily curved objects. If needed, the obtained w_{box} and h_{box} sizes can be increased by some margin d to yield boxes that encompass more context around each instance.

3.3 DETR for centerlines

We choose to base our approach on DETR for several reasons. Firstly, it is an end-to-end architecture with a simple implementation. In DETR, the notion of a bounding box only appears as the output of the model and in the loss formulation – that is in contrast to R-CNN-type methods, where the concepts of *proposals* and *anchors* are strictly tied to bounding box representation. That makes the reformulation of DETR for a different output representation much simpler.

Secondly, DETR’s attention-based detection scheme directly addresses the false detections problem. R-CNN methods require some form of Non-Maximum Suppression to reduce the large number of false positives - an operation that would need to be carefully defined for a centerline-based representation. In DETR however, only a single “detection slot” attends to each instance, while other slots do not duplicate their outputs, returning a **no-object** label instead.

We retain the entire architecture of DETR, with transformer encoder-decoder and learnable queries, only manipulating the output projection layer, changing the number of outputs from 4 to 6. The Hungarian algorithm-based matching scheme remains the same as well. However, we reformulate the *matching cost* function $\mathcal{L}(y_i, \hat{y}_j)$ between the ground truth y_i and a prediction with index j to compute similarity between predicted centerlines instead of bounding boxes.

By default, DETR defines its matching cost $\mathcal{L}_{\text{DETR}}$ as weighted similarity in class prediction and bounding box. Defining each ground truth instance as $y_i = (c_i, b_i)$, where c_i is the class label and $b_i \in [0, 1]^4$ is a vector of bounding box coordinates relative to the image space, and $\hat{p}_j(c_i)$ as the probability of class c_i for prediction of index j and $\hat{b}_j$ as the bounding box coordinates for this prediction, they obtain:

$$\mathcal{L}_{\text{DETR}}(y_i, \hat{y}_j) = -\log \hat{p}_j(c_i) + \mathbb{1}_{c_i \neq \emptyset} \mathcal{L}_{\text{box}}(b_i, \hat{b}_j) \quad (2)$$

Here, $\mathcal{L}_{\text{box}}$ is a weighted linear combination of L_1 loss over bounding box coordinates and the generalized IoU loss, introduced to smooth out the difference in scales of L_1 loss for small and large boxes of similar relative errors.

In our approach, we replace $\mathcal{L}_{\text{box}}$ with a similarity measure defined for centerlines, $\mathcal{L}_{\text{centerline}}$. We define it as a permutation-invariant L_1 loss over centerline coordinates. Let $r_i \in [0, 1]^6$ be a vector of centerline coordinates for a ground truth instance i , and $\hat{r}_j$ be a vector of centerline coordinates for prediction of index j , both relative to the image space. Let ρ be a function that permutes endpoint coordinates of a centerline:

$$\rho(x_c, y_c, x_{e_1}, y_{e_1}, x_{e_2}, y_{e_2}) = (x_c, y_c, x_{e_2}, y_{e_2}, x_{e_1}, y_{e_1}) \quad (3)$$

Then:

$$\mathcal{L}_{\text{centerline}} = \min(\|r_i - \hat{r}_j\|, \|r_i - \rho(\hat{r}_j)\|) \quad (4)$$

Considering both possible endpoint orderings and taking the smaller distance is necessary, as points on a 2D Cartesian plane (such as the image space) have no total ordering. In consequence, there is no function to define a “correct” order of endpoints that a model could learn. Therefore, both orderings must be considered valid.

Overall, we define our centerline matching loss $\mathcal{L}_{\text{CL}}$ as follows:

$$\begin{aligned} \mathcal{L}_{\text{CL}}(y_i, \hat{y}_j) &= -\log \hat{p}_j(c_i) + \mathbb{1}_{c_i \neq \emptyset} \lambda_{L1} \mathcal{L}_{\text{centerline}}(r_i, \hat{r}_j) \\ &= -\log \hat{p}_j(c_i) + \mathbb{1}_{c_i \neq \emptyset} \lambda_{L1} \min(\|r_i - \hat{r}_j\|, \|r_i - \rho(\hat{r}_j)\|) \end{aligned} \quad (5)$$

with λ_{L1} being a scalar weight, controlling the balance between spatial similarity and class probability losses. This parameter performs the same role as loss weights in DETR’s $\mathcal{L}_{\text{box}}$.

We omit the IoU loss component in our formulation. This decision has a mostly technical motivation: IoU is not a well-defined concept for centerlines. We believe it is not problematic, however, because most objects in our domain are of similar size, which eliminates the primary reason for introducing this loss in the first place.

3.4 Data

We train the proposed centerline model and the baseline on a dataset of 1500 images of dimensions $H = 1400\text{px}$, $W = 1600\text{px}$, containing a total of 135,000 mealworm instances with various levels of occlusion. This dataset was generated using computer graphics methods from a smaller pool of reference images with annotations, allowing the obtaining of a large number of highly varied images with annotations. The details of the generation procedure are discussed in our previous publication [3]. Data is generated already divided into training and validation splits, with 1200 training images (108,000 instances) and 300 validation images (27,000 instances).

Each instance in the original dataset was annotated with a class label, bounding box coordinates, and a dense segmentation mask. The mask was available in

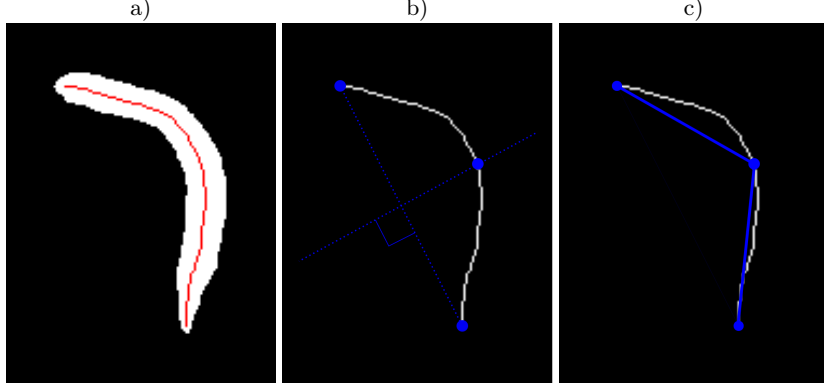


Fig. 3. Centerline extraction method. First, a binary segmentation mask is skeletonized (a). Next, skeleton endpoints are retrieved, and the section connecting them is bisected (b). Intersection of the perpendicular and the skeleton constitutes the midpoint (c).

two variants: an instance segmentation mask, marking each visible pixel belonging to an object, and an *amodal* segmentation mask, covering the entire extent of each instance, including the pixels occluded by other instances. Throughout the rest of paper whenever a segmentation mask is mentioned, it shall refer to the amodal variant.

In order to train the centerline-based detector, centerline ground truths need to be acquired. As explained in Section 3.2, this is generally not possible to do from bounding box annotation alone. Therefore we extract centerlines from the segmentation masks.

The procedure is explained in Figure 3. First, we perform skeletonization on the segmentation mask using `scikit-image` function `skeletonize` [17]. An example skeleton is marked in red on Figure 3a. Next, we extract endpoints of the skeleton, which will constitute endpoints of the centerline as well. Due to imperfect smoothness of the segmentation masks, in some cases skeletonization results in shapes that have more than two endpoints. For this reason, in the subsequent step we choose two maximally distant points from the set of skeleton endpoints.

To detect the center point of the skeleton, we construct a line segment connecting the identified endpoints, and find a perpendicular line bisecting it. Finally, we find the intersection of this line and the original skeleton, as can be seen in Figure 3b. This way we have constructed a 3-point representation that can be visually interpreted as two line segments - a centerline (Figure 3c).

3.5 Training

We construct our model following the reference DETR implementation, building a transformer with 6 encoder and 6 decoder layers of 8-head attention, and the hidden dimension 256. We retain the sinusoidal positional embedding, but change

the number of learnable queries to 300 (from the default 100) to account for the larger number of instances in our images. As the backbone, we adopt ResNet-50 [5], importing weights pretrained on ImageNet [2] from `torchvision` repository [12]. Due to the computational intensity of training a detection transformer from scratch, we also import weights for said transformer, pretrained on the MS COCO dataset [9].

For training the model, we use the same negative log likelihood loss for the classifier, and an L_1 -type loss for centerline predictor (see Equation 5). We set $\lambda_{L1} = 5$ to balance the contribution of the spatial loss. Following DETR, we attach auxiliary losses to each layer of the transformer decoder, and clip gradients greater than 0.1 in norm. We set the learning rate to 10^{-4} , except the backbone, which is trained at 10^{-5} to stabilize training in the early stages; weight decaying is set to 10^{-4} .

We train the model for 60,000 iterations (400 epochs, with 1,200 images per epoch and a batch size of 8). During training, we augment the images extensively using random flips in both dimensions and a custom random resizing strategy. Images are randomly upscaled, downscaled, or not rescaled at all, with equal probability. When rescaling, we vary the output aspect ratio by up to $\pm 10\%$, and limit the scale factor to $[0.8; 1.25]$ range. Finally, we crop or pad the resulting images as necessary to preserve the output shape of 1400x1600 pixels, in order to construct batches of uniform shape.

Both the centerline-based model and the reference bounding box model share the exact same hyperparameters and training scheme. We use a single Nvidia A40 GPU with 48 GB onboard memory, which requires approximately 18-20 hours to train each model.

3.6 Evaluation

Evaluating detection via segmentation: Quantitative comparison of the performance of the proposed model against references requires a common metric. That is not trivially possible given the difference in representation of the object in both approaches. The typical detection performance metric, mAP, relies on intersection over union (IoU) to declare predictions as correct or incorrect, given the IoU threshold. This notion is well-defined for a bounding box representation, but has no obvious analogue for centerlines. Instead of defining a corresponding “IoU-like” measure, which would only raise questions about the validity of its definition, we decide not to evaluate the detection performance directly.

Instead, we evaluate the models on a downstream task, choosing amodal instance segmentation for this purpose, because detection performance directly impacts segmentation performance. Amodal segmentation differs from the usual forms of semantic segmentation in that the model is asked to predict *all* pixels belonging to an object, not only the visible area. For both the proposed centerline-based DETR and the reference bounding-box-based DETR, we construct segmentation wrappers on top of the trained detectors, as per DETR design [1]. Additionally, we compare the results with a state-of-the-art amodal

instance segmentation model, AISFormer [16], which uses Mask R-CNN as a detector.

Naturally, the differences in segmentation quality need not be caused solely by strength differences of respective detectors. We take care to make the grounds for comparison as similar as possible. Both DETR-based segmentation models follow the same meta-architecture, with a simple convolutional segmentation head, and an attention-like conditioning layer. This conditioning layer, structured as simple attention without multiplication by value (only the activation softmax), provides the connection between the intermediate detection output from the transformer encoder and the segmentation head. The segmentation head follows an FPN approach, merging the conditioning information with the image features extracted by the final 3 blocks of the backbone network; it comprises 6 convolution layers with 3x3 kernels, ReLU activation and group normalization, using nearest-neighbor interpolation for upsampling the intermediate features to produce the final binary segmentation masks.

The segmentation models are trained using equally weighted Dice loss [11] and Focal loss [15]. The underlying detectors remain frozen, ensuring fairness of comparison. The same training hyperparameters are used: learning rate and weight decay of 10^{-4} , gradient norm clipping above 0.1, batch size 1. The models are trained for 72,000 iterations each (60 epochs, with 1,200 images per epoch), chosen to ensure convergence. With this procedure, we obtain segmentation models that can be evaluated on the validation set using the standard mAP measure.

Length estimation: To demonstrate the ability of the proposed method to retrieve useful geometry of the detected objects, we evaluate the performance of the detector on the task of mealworm length estimation. To this end, we extend the ground truth instances annotation with length values. Those values are acquired automatically at the centerline fitting stage (see Section 3.4), by computing the overall length of the skeleton.

We evaluate our model by measuring the length of the detected centerlines directly, i.e. without instance segmentation. The length is defined simply as the sum of the lengths of the two line segments constituting the centerline. After matching the detections to ground truths via bipartite matching (see Equation 5, Section 3.3), we compare the length of the predicted centerlines to the reference values. We compute root mean square error of the predictions and their relative error.

4 Experiments

4.1 Detection

Table 4.1 summarizes the results of segmentation performance evaluation on the validation dataset. Our centerline-based method outperforms baseline DETR by a small margin in terms of mean average precision, but when higher IoU thresholds are considered, e.g. $\text{IoU} = 0.75$ (AP75), the reference is slightly stronger.

Table 1. Performance of the segmentation models trained on top of the compared detectors. Transformer-based detectors achieve quality of predictions superior to Mask R-CNN. The proposed centerline-based approach further improves segmentation precision and recall.

Model	mAP	AP50	AP75	mAR
Mask R-CNN + AISFormer	0.716	0.910	0.879	0.752
DETR - bounding boxes	0.835	0.990	0.987	0.859
DETR - centerlines (ours)	0.842	0.990	0.983	0.869



Fig. 4. Example predictions of the centerline model on images from the validation set. In low-density scenarios (a), the detector performs very well even if the instances overlap slightly. With increasing object density (b), accuracy of endpoint placement decreases, and false positive predictions (duplicates) occasionally appear.

In terms of recall, our model offers a consistently higher recall rate of 0.869 (vs 0.859 by the baseline DETR). Segmentation model with underlying proposal-based detector, AISFormer, loses to fully transformer-based models significantly, regardless of their output representation.

The higher recall rate of the centerline-based model hints at its better ability to retrieve objects. This property would be desired especially in the context of amodal instance segmentation (AIS), where the occlusions between instances cause typical object detection methods to miss, resulting in false negative predictions. Our model achieves this improvement while simultaneously improving precision, making it even more attractive as a building block for AIS tasks.

Figure 4 shows example predictions of our model. In general, predicted centerlines fit the instances well, even if they are partially occluded. To some degree, the model is able to perform amodal completion of the instances; that means it correctly estimates the location of the endpoint of an object even in case it is not directly visible.

When density of the objects increases, some mistakes become visible. Primarily, the accuracy of endpoint coordinate regression begins to deteriorate. Most likely this is related to the limited resolution of the image tokens fed to the transformer encoder. Since the objects in the images are relatively small, the size of their fine details - such as endpoints - can be smaller than the token resolution.

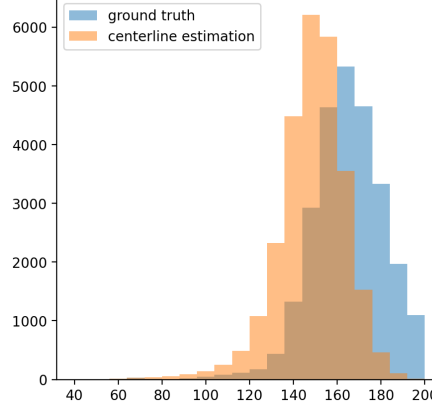


Fig. 5. The histogram of predicted object lengths versus the ground truth lengths. Centerlines are shorter, on average, than the true arcs of the object skeletons, causing the underestimation visible as a shift of the histogram.

When only a few objects are located in close proximity, tokens may encode those details in their representation, allowing the decoder to correctly infer the exact location of the endpoints. But with a tighter crowd, a single token might encompass a region corresponding to multiple instances, causing the representation to average over their features, thus hindering the detection quality.

4.2 Length estimation

Our model is capable of estimating the length of detected instances without the need to perform dense segmentation. We evaluate it by calculating the length of a detected centerline and matching it to a ground truth length obtained from the skeleton, obtaining $\text{RMSE} = 23.1$ (relative error of 14.0%). Figure 5 compares the histogram of estimated lengths with the reference histogram.

Due to the simplification of the object geometry, curvature will cause the predicted length to be smaller than the reference value. This is the direct cause of the visible shift of the predicted histogram to the left with respect to the ground truth. Simply scaling the predictions by the factor of 1.1 reduces this effect to some extent, causing the RMSE to drop to 17.0 and reducing the relative error to 10.3%.

More elaborate schemes, e.g. curve fitting, could be used to further improve those values. But even those results confirm that our model is able to extract meaningful and useful geometry directly during detection. This result would not be possible for a bounding-box-based representation at all, because the geometric information they encode is scarce.

5 Conclusion

We introduce a centerline-based representation for object detection and implement a transformer-based model for this representation, following DETR. Our representation provides a task-specific inductive bias, assuming that objects have an elongated shape. This allows us to perform object detection in a more geometrically meaningful way, that is permitting some shape analysis of the detected objects to be done without the need for instance segmentation.

Our method achieves strong results on our dataset, improving over the baseline bounding-box-based DETR in both precision and recall. The higher recall rate is a particularly attractive property in the context of amodal instance segmentation, where correctly retrieving occluded instances is challenging.

Future work should concern the underlying DETR framework, especially addressing its low spatial resolution, causing it to struggle with detecting very small objects and confusing boundaries between objects in close proximity.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. Ieee (2009)
3. Dolata, P., Majewski, P., Lampa, P., Zięba, M., Reiner, J.: Amodal instance segmentation for mealworm growth monitoring using synthetic training images, UNDER REVIEW
4. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2980–2988 (2017). <https://doi.org/10.1109/ICCV.2017.322>
5. 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. pp. 770–778 (2016)
6. Kirsten, L.N., Jung, C.R.: Cell tracking-by-detection using elliptical bounding boxes. arXiv preprint arXiv:2310.04895 (2023)
7. Liang, J., Homayounfar, N., Ma, W.C., Xiong, Y., Hu, R., Urtasun, R.: Poly-transform: Deep polygon transformer for instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9131–9140 (2020)
8. Lin, C., Li, C., Liu, Y., Chen, N., Choi, Y.K., Wang, W.: Point2skeleton: Learning skeletal representations from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4277–4286 (2021)

9. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
10. Majewski, P., Zapotoczny, P., Lampa, P., Burduk, R., Reiner, J.: Multipurpose monitoring system for edible insect breeding based on machine learning. *Scientific Reports* **12**(1), 7892 (2022)
11. Milletari, F., Navab, N., Ahmadi, S.A., Net, V.: Fully convolutional neural networks for volumetric medical image segmentation. In: Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV). pp. 565–571 (2019)
12. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimeshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E.Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. *CoRR abs/1912.01703* (2019), <http://arxiv.org/abs/1912.01703>
13. Redmon, J.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
14. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(6), 1137–1149 (2016)
15. Ross, T.Y., Dollár, G.: Focal loss for dense object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2980–2988 (2017)
16. Tran, M., Vo, K., Yamazaki, K., Fernandes, A., Kidd, M., Le, N.: Aisformer: Amodal instance segmentation with transformer. *arXiv preprint arXiv:2210.06323* (2022)
17. 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: scikit-image: image processing in Python. *PeerJ* **2**, e453 (6 2014). <https://doi.org/10.7717/peerj.453>, <https://doi.org/10.7717/peerj.453>