24th International Conference on Business Information Systems (BIS 2021)

Artificial Intelligence

https://doi.org/10.52825/bis.v1i.59



Published: 02 July 2021

# Evaluation of Deep Learning Instance Segmentation models for Pig Precision Livestock Farming

Jan-Hendrik Witte[1], Johann Gerberding[1], Christian Melching[1] and Jorge Marx Gómez[1]

[1] Universität Oldenburg, GER

**Abstract.** In this paper, the deep learning instance segmentation architectures DetectoRS, SOLOv2, DETR and Mask R-CNN were applied to data from the field of Pig Precision Livestock Farming to investigate whether these models can address the specific challenges of this domain. For this purpose, we created a custom dataset consisting of 731 images with high heterogeneity and high-quality segmentation masks. For evaluation, the standard metric for benchmarking instance segmentation models in computer vision, the mean average precision, was used. The results show that all tested models can be applied to the considered domain in terms of prediction accuracy. With a mAP of 0.848, DetectoRS achieves the best results on the test set, but is also the largest model with the greatest hardware requirements. It turns out that increasing model complexity and size does not have a large impact on prediction accuracy for instance segmentation of pigs. DETR, SOLOv2, and Mask R-CNN achieve similar results to DetectoRS with a parameter count almost three times smaller. Visual evaluation of predictions shows quality differences in terms of accuracy of segmentation masks. DetectoRS generates the best masks overall, while DETR has advantages in correctly segmenting the tail region. However, it can be observed that each of the tested models has problems in assigning segmentation masks correctly once a pig is overlapped. The results demonstrate the potential of deep learning instance segmentation models in Pig Precision Livestock Farming and lay the foundation for future research in this area.

**Keywords:** Precision Livestock Farming, Instance Segmentation, Computer Vision, Deep Learning, Pig.

## Introduction

Structures of modern pig livestock farming, and pork production have been undergoing major changes in recent years. Data from the Federal Statistical Office show the opposite trend of a steadily decreasing number of farms [1] with simultaneously increasing numbers of animals per farm [2] and a continuously decreasing slaughter price[1], which poses and will continue to pose great challenges for the farmer. At the same time, politics and society alike are calling for more sustainable and more animal-friendly husbandry [3], which puts additional pressure on the farmer and makes economically profitable pig livestock farming increasingly difficult. These challenges cannot be met with conventional methods, which is why new and innovative solutions are needed. As a result, research in the domain of Precision Livestock Farming (PLF) has increased in recent years. PLF describes systems that utilizes modern camera and sensor technologies to enable automatic real-time monitoring in livestock production to supervise animal health, welfare and behaviour [3], [4]. This involves the automated acquisition, processing, analysis and evaluation of sensor-based data like temperature, humidity or CO2-concentration [5] and image and video data [6], [7]. Based on

[1] https://www.bmel-statistik.de/preise/preise-fleisch/

this information, systems can be created that support the farmer in his daily work and help him adapt to the changing conditions in pig livestock farming. To enable such systems, methods are first needed that allow the automated processing of these different data streams in the form of image, video, and sensor data. In the case of image data, methods are required that can be used for automated recognition and localization of individual pigs within

the pen. Due to the specific conditions in pigsties, these tasks pose a particular challenge.

**Figure 1**. Example image from pigsty.

Fig. 1 illustrates some of these problems. Piling, crowded areas, the overlapping of pigs as well as their various orientations and alignments make it difficult to automatically detect and locate individual pigs within the pen. In addition, there are constantly changing factors such as varying light conditions, soiling of the animals and the pen and occlusions caused by objects in the pen. In literature, similar use cases such as automated pedestrian detection in crowded areas have been successfully addressed using deep learning (DL) methods [8]. For pig detection and localisation, there are two approaches that are used in DL domain: (1) Object detection and (2) instance segmentation. Object detection (1) describes the classification and localisation of objects with the help of bounding boxes [9]. The algorithm surrounds each classified object within the image with a bounding box, which can then be used to determine the object's position in the image. Instance segmentation is a combination of object recognition and semantic segmentation. After object detection, semantic segmentation is used to classify and map each pixel of the detected object to a corresponding class or category. This results in detailed masks for each detected object where each mask can be considered as an independent instance [9].

A variety of different object detection methods from the field of DL have already been used in PLF. Cang et al. applied Faster-RCNN for detection and weight estimation of pigs based on image data [10], Nasirahmadi et al. utilized Single Shot Multibox detector, Region-Based Fully Convolutional Neural Network and Faster R-CNN for posture detection of individual pigs [11] and YOLO was applied by Sa et al. for pig detection under various illumination conditions [12]. However, bounding boxes are not able to capture the contours of objects, which is why valuable information could be lost when only using bounding boxes [13]. For some PLF related use cases like the prediction of tail biting in grouped house pigs, this information could be insufficient. In a report by the BMEL[2], in which various indicators for the early detection of tail biting were summarised, it emerges that activities such as tail-in-mouth behaviour or generally manipulative chewing behaviour on pen objects can increasingly be observed before tail biting events [14]. For this type of use case, the much more precise instance segmentation masks could be beneficial. In the DigiSchwein[3] project, the automated early detection of tail biting is a central objective, which we intend to explore

[2] Bundesministerium für Ernährung und Landwirtschaft

[3] https://www.lwk-niedersachsen.de/index.cfm/portal/1/nav/1093/article/35309.html

further with the help of modern deep learning methods. For this reason, this paper investigates whether common instance segmentation methods from the field of DL can be applied to data from Pig PLF. Using defined selection criteria, four different instance segmentation methods are identified in DL literature and tested and evaluated based on a custom dataset. The goal is to evaluate whether the applied methods can deal with the specific challenges of the data from the Pig PLF domain and how their quality is in terms of prediction accuracy and speed.

This paper is structured as follows. First, we examine how instance segmentation has been applied in the context of Pig PLF and which methods have been used. Based on these results, the selection criteria for the instance segmentation methods are presented, followed by a brief description of the selected models. The presentation of the results is done using a quantitative and qualitative analysis. The model evaluation is performed using the mean average precision (mAP) based on our test set that we extracted from our annotated dataset. The qualitative analysis is based on a visual evaluation of the predicted masks by the different models and is used to discuss potential problems and remaining challenges. The interpretation of the results discusses the insights gained from the quantitative and qualitative evaluation. The conclusion and outlook summarize the results and describe how they can be used in future research.

## Related Work

Instance segmentation use cases identified in literature can be divided into two different categories: (1) segmentation without DL and (2) segmentation with DL [15]. Segmentation techniques without DL (1) are characterised by using thresholding for image binarization, separating background from foreground [16]. Otsu's method is a popular example for this, which has been used in a variety of use cases [17], [18], [19], [20]. This type of segmentation is usually only done as a pre-processing step, based on which the actual detection of the objects in the foreground takes place. Nasirahmadi et al. use the results of Otsu's binarization to locate pigs on image data using an ellipse fitting algorithm [21]. The approach for the identification and localisation of grouped-house pigs by Huang et al. has a similar structure, but Gabor filters are used for segmentation and feature extraction. The subsequent detection is done with Support Vector Machines [22]. However, methods like Otsu's do not perform instance segmentation, but semantic segmentation since the entire content of an image is segmented according to the defined threshold rather than individual regions or instances. Due to the definition of a threshold for image segmentation, such solutions are also vulnerable to structural changes within the image like changing light conditions, occlusion or dirt [20]. To address these problems and enable actual instance segmentation of objects, DL methods can be applied.

During literature research, only three papers were identified that applied instance segmentation methods from the field of DL to Pig PLF. Seo et al. conclude that the predictions of the Mask R-CNN are insufficient for the use case of separating touching pigs in image data. They describe that the segmentation accuracy of the predicted masks is not satisfactory, as some pigs in overcrowded areas are not recognised correctly or are completely missed out [23]. On the other hand, Li et al. successfully use Mask R-CNN for instance segmentation of pigs. They use the information provided by the segmentation masks to automatically recognise mounting behaviour. The presented model was fine-tuned on pre-trained COCO model and a ResNet50 backbone [24]. Tu et al. tested and evaluated Mask Scoring R-CNN, an adaptation of Mask R-CNN, to improve instance segmentation performance for grouped-housed pigs [15]. Mask Scoring R-CNN improves instance segmentation performance by adding a network block that learns the quality of the predicted instance masks and feeds this information back into the network during training [25].

During analysis of these papers, we noticed some problems in the way the evaluation of instance segmentation models was conducted. In research, the COCO data format has become the standard format to train and evaluate instance segmentation models [26], [27].

Most of the instance segmentation models and methods found in literature, including Mask R-CNN, use the COCO evaluation format to benchmark model performance against other architectures [9]. The commonly used metric for evaluation and benchmarking is the mAP, which is the mean of the Average Precision (AP) based on a set of different Intersection over Union (IoU) thresholds [9]. IoU is the most used evaluation metric for object detection and instance segmentation tasks. It is defined as the similarity between the ground truth segmentation and the predicted segmentation present in the image and is determined by dividing the intersection with the area of union [28]. We noticed that none of the identified paper uses this metric to evaluate their model. Tu et al. use Precision, Recall and F1-Score for evaluation [15], Seo at al. did not state a general model performance for instance segmentation at all [23] and Li et al. used mean Pixel Accuracy (mPA) metric [24] which is normally used to evaluate performance in semantic segmentation tasks and not instance segmentation [29]. Pixel accuracy describes the amount or percentage of pixels which are classified correctly by the model. This can be problematic if, for example, there is a class imbalance in the data used, in which the number of pixels in the image that do not belong to any class greatly exceeds the number of pixels that belong to a class and vice versa, thus enabling a classification performance based on the class imbalance with a priori knowledge [29]. This results in two problems: (1) A comparison of the performance of the respective results is not possible due to the different and partly inappropriate evaluation metrics and (2) No evaluation of instance segmentation models on data from the Pig PLF domain has been conducted yet based on the standard evaluation metric mAP. These research gaps are also addressed in this paper.

# Materials and methods

## Instance segmentation model selection

Instance segmentation methods were chosen based on defined selection criteria. The definition of the selection criteria was made based on different aspects. On the one hand, the requirements for PLF systems mentioned in the literature should be considered. On the other hand, the selection criteria should serve to answer the research question of this paper regarding prediction accuracy and speed of the model on data from the Pig PLF domain. The following criteria were defined:

1. **Prediction Accuracy**: The prediction of the respective model should be as accurate as possible [30].
2. **Prediction Speed**: Model inference should be in real time [18].

Prediction speed in real time refers to the requirement that the respective algorithm should deliver a result within milliseconds. Cost-effectiveness is a criteria that is mentioned in literature as well regarding PLF systems [18] but has been ignored in the context of this paper as it does not contribute to answering the initial research question. To set a baseline for which models to compare and to allow consideration of innovative approaches for instance segmentation, two additional selection criteria were added:

3. **Innovation**: Architectures with new or innovative approaches are to be examined for suitability.
4. **Usage in PLF research**: The recently used instance segmentation architectures in PLF literature should be considered for comparison.

The website paperswithcode[4] provides an overview of all published instance segmentation architectures and their benchmark results on the COCO test-dev, a dataset on which model performance is evaluated and benchmarked. Since we also use the COCO format for training and evaluation in this paper, this overview serves as a basis for selecting the instance segmentation models based on our criteria. For each defined criterion, an instance

[4] https://paperswithcode.com/task/instance-segmentation

segmentation model was chosen. The following models were selected and evaluated in this paper:

**DetectoRS**: DetectoRS achieves state of the art (SOTA) performance on COCO test-dev for instance segmentation [31], which is why it was selected for this paper based on criterion 1. Inspired by the human mechanism of looking and thinking twice, the authors tried to implement similar mechanisms into their architecture at both the macro and the micro level. At the macro level, a Recursive Feature Pyramid (RFP) is proposed which builds on top of a Feature Pyramid Network (FPN). The RFP incorporates additional feedback connections from the FPN layers into the bottom-up backbone layers which creates a recursive operation. At the micro level, Switchable Atrous Convolutions (SAC) are incorporated, which convolves the same input feature with different atrous rates and gathers the results using switch functions. The incorporation of this mechanism into both levels improved the mAP on COCO test-dev by up to 4.3%.

**SOLOv2**: SOLO is an anchorless instance segmentation approach introduced by Wang et al. [32]. It divides an image into a uniform grid with each grid cell being responsible for the detection of an object if its centre is placed in it. Class probabilities and a global binary mask are computed for each cell individually. This restricts each cell to only predict one object or class. The model combines a FPN with a category and mask branch and reduces the problem of instance segmentation to the question of which cell and category a pixel belongs to. SOLOv2 extends this idea by improving the mask branch of the model [33]. Two new branches are introduced, the kernel branch and the feature branch. The kernel branch is responsible for generating a kernel for each cell of the grid while the feature branch generates multiple different prototype masks. Finally, the kernels are used in a convolution operation on the prototype masks to generate the final predictions. SOLOv2 archives SOTA performance in real time inference and is selected based on criteria 2.

**DETR**: DETR describes a novel object detection method introduced by Carion et al. [34]. The method offers a new approach to object detection as it uses a feature extractor in combination with Transformers [35]. The model uses the feature vectors extracted by a Convolutional Neural Network (CNN) backbone as an input for the encoder and its attention heads. The decoder then generates a defined number of predictions in parallel, each of which is assigned to a class either from the given dataset or an additional class like the background class. While its initial implementation focusses on the prediction of bounding boxes, the authors also demonstrate the adaptation to instance segmentation. This is done by utilizing the output of the decoders in combination with the multi-head-attention values und multiple convolutional layers to generate upscaled binary masks. Due to the innovative approach of instance segmentation with transformers, DETR was selected based on criteria 3.

**Mask R-CNN**: Mask R-CNN extends on Faster R-CNN by adding an additional branch for masks prediction that works parallel to the branch for bounding box prediction. Thus, the pixel wise prediction of the individual segmentation masks is decoupled from the actual classification of the object. A RoIAlign layer is introduced to improve bounding box alignment after RoIPooling and preserves exact spatial locations. For segmentation mask prediction, a Fully Convolutional Network (FCN) is used on each of the extracted bounding boxes [9]. Since Mask R-CNN is the most current instance segmentation architecture in PLF literature and no evaluation in COCO format has been performed yet, Mask R-CNN was selected based on criteria 4.

## Dataset description

To evaluate the performance of selected instance segmentation architectures, a dataset consisting of a total of 731 images with high quality segmentation masks was created. The open source tool Labelme was used to annotate the images and convert them into the COCO format [36]. To ensure heterogeneity in the data, the dataset was compiled from a combination of samples from several datasets. Fig. 2 shows exemplary images that

illustrates the data heterogeneity. When creating the dataset, we tried to cover as many different backgrounds, camera perspectives, shooting lenses and lighting conditions as possible. Care was also taken to include PLF specific challenges such as piling, occlusion or overlapping of pigs in the dataset. Psota et al. published a dataset of a total of 2000 keypoint

**Figure 2**. Images from the dataset

**Figure 3**. Mask example

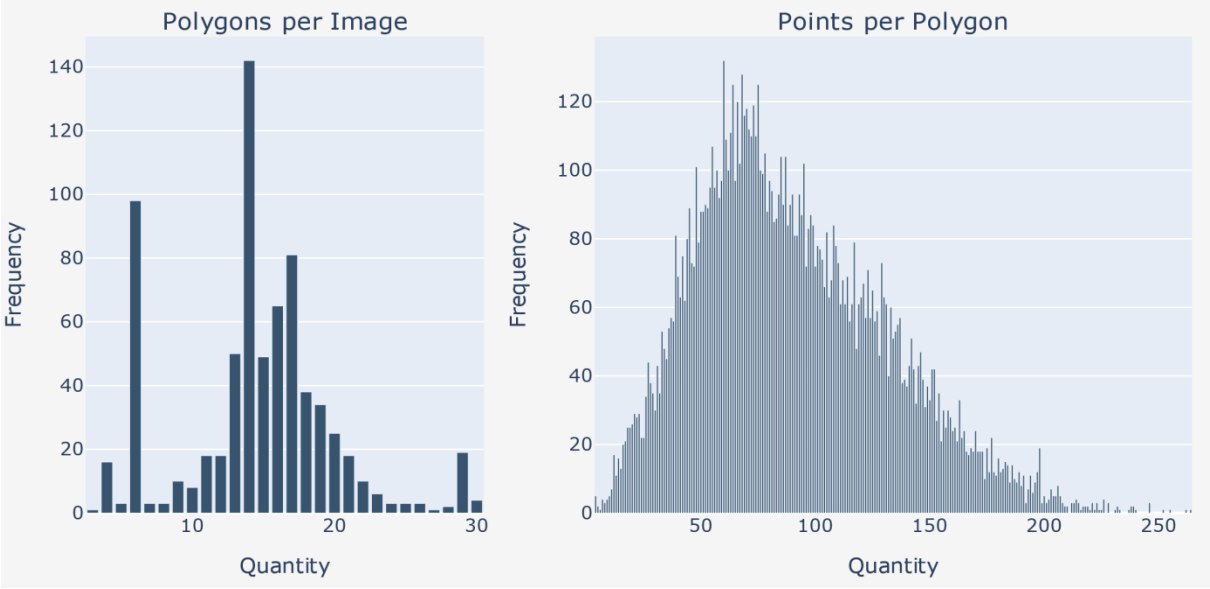


**Figure 4**. Descriptive statistics of the dataset.

annotated images from 17 different locations [37], of which a total of 631 images were extracted and annotated. The remaining 94 images were provided by the KoVeSch[5] project of the Lower Saxony Chamber of Agriculture, which includes 60 pictures from piglet rearing and 34 pictures with fattening pigs. All images were annotated by hand. Three people were involved in the labelling process, with each annotated image being checked by another person to assure quality and correctness of the annotations. Fig. 4 shows some descriptive statistics about the dataset. Each image contains between 3 and 30 pigs, while the average number of pigs per image is 14.5. The average number of coordinate points per mask is about 90, while a few masks can consist of up to 264 points. Fig. 3 shows an example of

[5] https://www.lwk-niedersachsen.de/index.cfm/portal/1/nav/1093/article/34849.html

annotated masks. Compared to the other datasets, our dataset has a higher number of pigs per image and a higher variation of data within the set, as both different locations and different camera angles were considered. The data set was divided into a training and test set, with 75% being used for training and 25% being used for testing and benchmarking.

### Test environment and setup

Model training was performed on a desktop workstation with two Nvidia RTX 3090 with 24 GB VRAM each, a Threadripper 3960X and 64 GB RAM. The MMDetection framework was used to train and evaluate Mask-RCNN and DetectoRS [38], while the AdelaiDet toolbox was used for SOLOv2 [39]. As the DETR implementation in MMDetection did not provide instance segmentation, the original implementation of the authors[6] was used instead. To check the suitability of the models for instance segmentation in the field of Pig PLF, it was decided to use the default configuration for each model and to not make any adjustments to the parameters. For each training job, we fine-tuned the respective architecture using models pre-trained on the COCO dataset. A Resnet50 backbone was used for each of the tested models, so that, apart from a few deviations in the respective configuration file, all models were trained and evaluated on the same baseline. Each model was fine-tuned over 30 epochs.

## Results

### Quantitative evaluation

The models were evaluated based on their mAP including $AP^{IoU=0.50}$ and $AP^{IoU=0.75}$, inference speed on GPU and CPU, and number of parameters. The number of parameters describes the model size and can affect the required hardware to train and operationalize the respective model. The IoU threshold specifies the threshold at which a prediction is classified as true positive, while the mAP represents the average of all determined APs. The mAP was calculated from the results for IoU thresholds in the range 0.5 to 0.95 with a step size of 0.05 represented as AP@[.5:.05:.95] [27].

**Table 1.** Results of the evaluation on the test set.

| Model | Average precision | | | Inference time in s | | # Parameters |
|---|---|---|---|---|---|---|
| | mAP | $AP^{IoU=0.50}$ | $AP^{IoU=0.75}$ | GPU | CPU | |
| DetectoRS | **0.848** | 0.978 | **0.947** | 0.147 | - | 131,648,615 |
| SOLOv2 | 0.831 | **0.980** | 0.946 | 0.097 | **0.905** | 46,175,681 |
| DETR | 0.830 | 0.976 | 0.933 | 0.122 | 2.262 | **42,613,152** |
| Mask R-CNN | 0.822 | 0.978 | 0.946 | **0.066** | 1.574 | 43.971,158 |

As seen in Tab. 1, the best performance in prediction accuracy was achieved using DetectoRS. The model achieves a mAP of 0.848 and is thus slightly better than the competition, but also has the highest resource requirements. With 131 million parameters, DetectoRS is by far the biggest model compared to the others, although inference on GPU is only slightly slower than the more lightweight SOLOv2. For DetectoRS, inference on CPU was not possible because it was not supported by the used framework. On GPU, Mask R-CNN was the fastest among the tested models with an inference time of 0.06s per image, but provides the lowest mAP compared to the other models. However, measured by $AP^{IoU=0.50}$, it can be observed that Mask R-CNN gives a better result than DETR and achieves identical performance to DetectoRS. This is also true for $AP^{IoU=0.75}$, although the difference between Mask R-CNN and DETR is even greater here. SOLOv2 is slightly slower than Mask R-CNN

[6] https://github.com/facebookresearch/detr

but has the best overall result for $AP^{IoU=0.50}$. It is also the fastest model when tested on CPU. DETR represents the smallest model with a total of 42 million parameters but is also the slowest on CPU. Overall, SOLOv2, DETR, and Mask R-CNN do not differ significantly in their parameter size. In general, it is noticeable that the results for $AP^{IoU=0.50}$ of all tested models are similar. Differences in performance are only noticeable at higher thresholds.

## Qualitative evaluation

For the qualitative evaluation of the predicted masks, we selected an example image from the test set that included as many of the mentioned visual challenges in Pig PLF as possible. For visualization purposes, the visualization method provided by the respective package was used, which explains the different coloured masks of the respective visualizations. Fig. 5 shows the predictions of the different models on the selected example image. As with the measured mAPs, the quality of the predicted masks is at a similar level for all models. Different coloured points were placed on the image to mark specific areas where the quality of the predicted masks sometimes deviated significantly. The red points indicate that in some cases, DETR and SOLOv2 have problems assigning pixels to the correct instance. While DETR does not assign the pixels to any instance at all, SOLOv2 tends to assign them to the wrong one. However, these errors can be observed in any of the tested architectures. This inconsistency can be illustrated with the help of the white points that focus on the pig's tail.


**Figure 5**. Predictions on test image.

Although DetectoRS produces the highest overall quality masks in comparison, correct segmentation of the tail is possible in only 2 out of 5 cases. In this task, DETR generates the best segmentations for the tail region. A problem that every tested model struggles with is highlighted by the orange dots. Due to the overlap of the pigs, the legs cannot be assigned to the correct mask by any model. The weaknesses of the models highlighted by the different points can be found consistently in other test images as well.

After comparing the individual models and identifying the best model based on mAP and visual evaluation, the generalization ability of the model should also be demonstrated. For this purpose, a series of images was selected from the test set containing images with as many different camera positions, shooting angles, camera lenses and light conditions as possible. Fig. 6 shows an overview of these images. As can be seen in the images, the segmentation of the individual masks also succeeds in completely different scenarios.

Although there are similar detail errors as in Fig. 5, the general quality of the masks is already promising. Of all the images, there is only one in which a pig was not provided with a mask, found in red rectangle. This problem occurred in about 10% of the tested images and if present, always occurs in crowded areas. This is similar to the problem mentioned by Seo et al. and is caused by the Non-Maximum Suppression (NMS) algorithm, which He et al. also pointed out in their introduction of the Mask-R CNN architecture [9]. Adjusting the threshold of the NMS could therefore lead to better results. It is also be seen that in theory the model can be applied in different pig compartments as well. Both piglets and fattening pigs can be correctly segmented by the model.

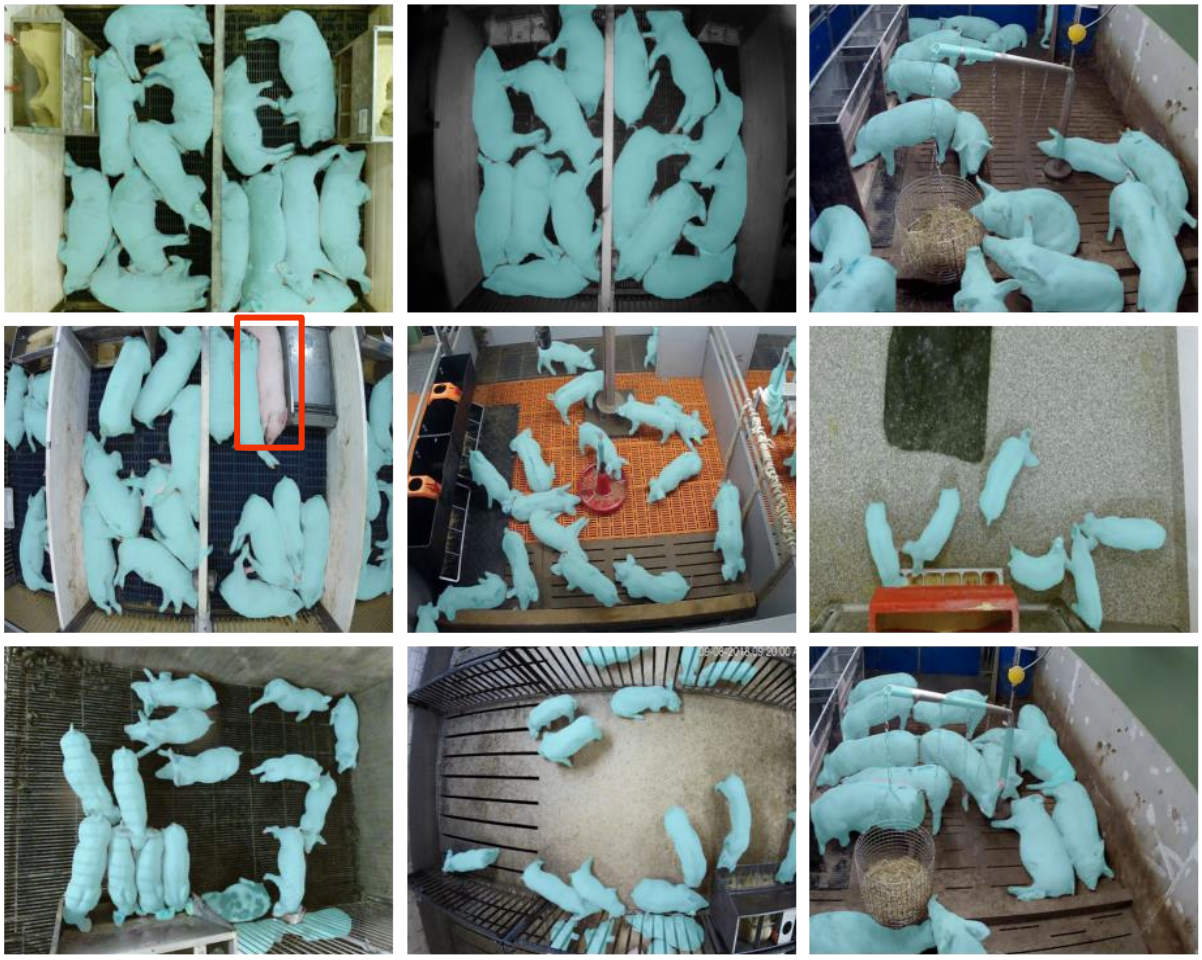

**Figure 6**. Predictions of DetectoRS.

## Interpretation of the results

Both the quantitative and the qualitative evaluation show that the results of the models do not differ significantly in terms of prediction accuracy. In this context, it can also be stated that each of the four instance segmentation models we tested is applicable to Pig PLF data when it comes to prediction accuracy. The fact that the prediction accuracy of the models is so close may be related to the fact that in this case of the Pig PLF, only one class needs to be predicted, whereas the COCO dataset on which the models evaluated here were trained on, contains 80 different classes. Although the number of parameters of DetectoRS is almost three times larger than that of the other models, it does not result in a significantly better mAP. This means that increasing the size and complexity of the model does not necessarily have a significant effect on improving the mAP in this type of use case. Accordingly, the problems identified in the qualitative evaluation for instance segmentation of pigs cannot be solved by scaling the models vertically in depth but require other approaches to improve segmentation accuracy. Taking the prediction speed and the number of parameters into account, it can be stated that SoloV2, DETR and MASK R-CNN are better suited than

DetectoRS for potential use cases in this domain under current circumstances. This is due to the fact that fewer resources are required to operationalize these models, allowing them to be deployed on less expensive hardware. With respect to the prediction accuracy, it is also necessary to differentiate for which use case the respective models should be used and what the requirements for accuracy are in this case. Based on the results, it can be stated that Mask R-CNN can be used as a baseline due to its fast execution time, small size, and its accuracy. For use cases such as tail bite detection, where more precise masks might be needed, DETR could be used as a baseline.

## Conclusion and outlook

In this paper, we demonstrated the usability of the deep learning instance segmentation models DetectoRS, SOLOv2, DETR and Mask R-CNN when being applied to data from the field of PLF for instance segmentation of pigs. The standard evaluation metric mAP was also applied for the first time to uniformly evaluate deep learning instance segmentation models in the Pig PLF domain to make performance more comparable. The results show that, in terms of prediction accuracy, each of the tested models can in principle be applied for instance segmentation of pigs. We observed that for instance segmentation in this context, the complexity and size of the model does not have a significant impact on the mAP, as the less complex models Mask R-CNN, SOLOv2 and DETR achieve similar prediction accuracy compared to DetectoRS. Accordingly, the identified problems in pig instance segmentation such as incorrect mask assignment when pigs overlap cannot be solved by vertically scaling models in depth but require other approaches or improvements. Based on this work, future research in this domain will focus on the aspect of cost efficiency when evaluating instance segmentation models for PLF systems. For example, it could be investigated which of the tested models can be deployed and operationalized on low-cost hardware or edge devices. Since in the context of this paper only the default configuration of each framework was applied to create the training jobs, the optimization of the configuration parameters could also be a direction for future research. Here, methods such as hyperparameter tuning could be applied to find an optimal configuration of the respective models to investigate the influence of these on the mAP. Alternative instance segmentation models and architectures such as YOLACT could also be explored in this domain for suitability in future research.

## References

[1] Statisitsches Bundesamt (Destatis), *Betriebe: Deutschland, Jahre, Tierarten.* [Online]. Available: https://www-genesis.destatis.de/genesis/online, Code: 41311-0003 (accessed: Feb. 17 2021).

[2] Statisitsches Bundesamt (Destatis), *Gehaltene Tiere: Deutschland, Jahre, Tierarten.* [Online]. Available: https://www-genesis.destatis.de/genesis/online, Code: 41311-0001 (accessed: Feb. 17 2021).

[3] D. Berckmans, “Precision livestock farming technologies for welfare management in intensive livestock systems,” *Revue scientifique et technique (International Office of Epizootics)*, vol. 33, no. 1, pp. 189–196, 2014, doi: 10.20506/rst.33.1.2273.

[4] R. B. D'Eath *et al.,* “Automatic early warning of tail biting in pigs: 3D cameras can detect lowered tail posture before an outbreak,” *PloS one*, vol. 13, no. 4, e0194524, 2018, doi: 10.1371/journal.pone.0194524.

[5] J. Cowton, I. Kyriazakis, T. Plötz, and J. Bacardit, “A Combined Deep Learning GRU-Autoencoder for the Early Detection of Respiratory Disease in Pigs Using Multiple Environmental Sensors,” *Sensors (Basel, Switzerland)*, vol. 18, no. 8, p. 2521, 2018, doi: 10.3390/s18082521.

[6] C. Chen *et al.,* “Recognition of aggressive episodes of pigs based on convolutional neural network and long short-term memory,” *Computers and Electronics in Agriculture*, vol. 169, p. 105166, 2020, doi: 10.1016/j.compag.2019.105166.

[7] C. Chijioke Ojukwu, Y. Feng, G. Jia, H. Zhao, and H. Ta, “Development of a computer vision system to detect inactivity in group-housed pigs,” *International Journal of Agricultural and Biological Engineering*, vol. 13, no. 1, pp. 42–46, 2020, doi: 10.25165/j.ijabe.20201301.5030.

[8] S. Zhang, J. Yang, and B. Schiele, “Occluded Pedestrian Detection Through Guided Attention in CNNs,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition: CVPR 2018 : proceedings : 18-22 June 2018, Salt Lake City, Utah*, Salt Lake City, UT, 2018, pp. 6995–7003, doi: 10.1109/CVPR.2018.00731

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” Mar. 2017. Accessed: May 28 2020. [Online]. Available: http://arxiv.org/pdf/1703.06870v3

[10] Y. Cang, H. He, and Y. Qiao, “An Intelligent Pig Weights Estimate Method Based on Deep Learning in Sow Stall Environments,” *IEEE Access*, vol. 7, no. 99, pp. 164867–164875, 2019, doi: 10.1109/ACCESS.2019.2953099.

[11] A. Nasirahmadi *et al.,* “Deep Learning and Machine Vision Approaches for Posture Detection of Individual Pigs,” *Sensors (Basel, Switzerland)*, vol. 19, no. 17, 2019, doi: 10.3390/s19173738.

[12] J. Sa, Y. Choi, H. Lee, Y. Chung, D. Park, and J. Cho, “Fast Pig Detection with a Top-View Camera under Various Illumination Conditions,” *Symmetry*, vol. 11, no. 2, p. 266, 2019, doi: 10.3390/sym11020266.

[13] S. Küster, M. Kardel, S. Ammer, J. Brünger, R. Koch, and I. Traulsen, “Usage of computer vision analysis for automatic detection of activity changes in sows during final gestation,” *Computers and Electronics in Agriculture*, vol. 169, p. 105177, 2020, doi: 10.1016/j.compag.2019.105177.

[14] S. Schukat and H. Heise, “Indikatoren für die Früherkennung von Schwanzbeißen bei Schweinen – eine Metaanalyse,” (in de) *Berichte über Landwirtschaft - Zeitschrift für Agrarpolitik und Landwirtschaft*, vol. 11, no, 22, 2019, doi: 10.12767/BUEL.V97I3.249.

[15] S. Tu *et al.,* “Instance Segmentation Based on Mask Scoring R-CNN for Group-housed Pigs,” in *2020 International Conference on Computer Engineering and Application: ICCEA 2020 : 27-29 March 2020, Guangzhou, China : proceedings*, Guangzhou, China, 2020, pp. 458–462, doi: 10.1109/ICCEA50009.2020.00105

[16] B. Li, L. Liu, M. Shen, Y. Sun, and M. Lu, “Group-housed pig detection in video surveillance of overhead views using multi-feature template matching,” *Biosystems Engineering*, vol. 181, pp. 28–39, 2019, doi: 10.1016/j.biosystemseng.2019.02.018.

[17] A. Nasirahmadi, S. A. Edwards, S. M. Matheson, and B. Sturm, “Using automated image analysis in pig behavioural research: Assessment of the influence of enrichment substrate provision on lying behaviour,” *Applied Animal Behaviour Science*, vol. 196, pp. 30–35, 2017, doi: 10.1016/j.applanim.2017.06.015.

[18] S. Lee, H. Ahn, J. Seo, Y. Chung, D. Park, and S. Pan, “Practical Monitoring of Undergrown Pigs for IoT-Based Large-Scale Smart Farm,” *IEEE Access*, vol. 7, pp. 173796–173810, 2019, doi: 10.1109/ACCESS.2019.2955761.

[19] J. Kim *et al.,* “Depth-Based Detection of Standing-Pigs in Moving Noise Environments,” *Sensors (Basel, Switzerland)*, vol. 17, no. 12, 2017, doi: 10.3390/s17122757.

[20] K. Jun, S. J. Kim, and H. W. Ji, “Estimating pig weights from images without constraint on posture and illumination,” *Computers and Electronics in Agriculture*, vol. 153, pp. 169–176, 2018, doi: 10.1016/j.compag.2018.08.006.

[21] A. Nasirahmadi, O. Hensel, S. A. Edwards, and B. Sturm, “Automatic detection of mounting behaviours among pigs using image analysis,” *Computers and Electronics in Agriculture*, vol. 124, pp. 295–302, 2016, doi: 10.1016/j.compag.2016.04.022.

[22] W. Huang, W. Zhu, C. Ma, Y. Guo, and C. Chen, “Identification of group-housed pigs based on Gabor and Local Binary Pattern features,” *Biosystems Engineering*, vol. 166, pp. 90–100, 2018, doi: 10.1016/j.biosystemseng.2017.11.007.

[23] J. Seo, J. Sa, Y. Choi, Y. Chung, D. Park, and H. Kim, “A YOLO-based Separation of Touching-Pigs for Smart Pig Farm Applications,” in *2019 21st International Conference on Advanced Communication Technology (ICACT)*, 2019, pp. 395–401, doi: 10.23919/ICACT.2019.8701968.

[24] D. Li, Y. Chen, K. Zhang, and Z. Li, “Mounting Behaviour Recognition for Pigs Based on Deep Learning,” *Sensors (Basel, Switzerland)*, vol. 19, no. 22, 2019, doi: 10.3390/s19224924.

[25] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, “Mask Scoring R-CNN,” Mar. 2019. [Online]. Available: https://arxiv.org/pdf/1903.00241

[26] T.-Y. Lin *et al.,* “Microsoft COCO: Common Objects in Context,” May. 2014. [Online]. Available: https://arxiv.org/pdf/1405.0312

[27] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, “A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit,” *Electronics*, vol. 10, no. 3, p. 279, 2021, doi: 10.3390/electronics10030279.

[28] M. A. Rahman and Y. Wang, “Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation,” in *ISVC*, 2016, doi: 10.1007/978-3-319-50835-1_22

[29] M. Thoma, “A Survey of Semantic Segmentation,” Feb. 2016. [Online]. Available: https://arxiv.org/pdf/1602.06541

[30] T. Norton, C. Chen, M. L. V. Larsen, and D. Berckmans, “Review: Precision livestock farming: building 'digital representations' to bring the animals closer to the farmer,” *animal*, vol. 13, no. 12, pp. 3009–3017, 2019, doi: 10.1017/S175173111900199X.

[31] S. Qiao, L.-C. Chen, and A. Yuille, “DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution,” Jun. 2020. [Online]. Available: https://arxiv.org/pdf/2006.02334

[32] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, “SOLO: Segmenting Objects by Locations,” Dec. 2019. [Online]. Available: https://arxiv.org/pdf/1912.04488

[33] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, “SOLOv2: Dynamic and Fast Instance Segmentation,” Mar. 2020. [Online]. Available: https://arxiv.org/pdf/2003.10152

[34] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-End Object Detection with Transformers,” May. 2020. Accessed: May 28 2020. [Online]. Available: http://arxiv.org/pdf/2005.12872v2

[35] A. Vaswani *et al.,* “Attention Is All You Need,” Jun. 2017. [Online]. Available: https://arxiv.org/pdf/1706.03762

[36] Kentaro Wada, *labelme: Image Polygonal Annotation with Python*.

[37] E. T. Psota, M. Mittek, L. C. Pérez, T. Schmidt, and B. Mote, “Multi-Pig Part Detection and Association with a Fully-Convolutional Network,” *Sensors (Basel, Switzerland)*, vol. 19, no. 4, p. 852, 2019, doi: 10.3390/s19040852.

[38] K. Chen *et al.,* “MMDetection: Open MMLab Detection Toolbox and Benchmark,” *arXiv preprint arXiv:1906.07155*, 2019.

[39] Z. Tian, H. Chen, X. Wang, Y. Liu, and C. Shen, *AdelaiDet: A Toolbox for Instance-level Recognition Tasks*.