Using Deep Learning for Automated Tail Posture Detection of Pigs

Jan-Hendrik Witte
Very Large Business Applications
University of Oldenburg
Oldenburg, Germany
jan-hendrik.witte@uni-oldenburg.de

Johann Gerberding
Very Large Business Applications
University of Oldenburg
Oldenburg, Germany
johann.gerberding@uni-oldenburg.de

Jorge Marx Gómez
Very Large Business Applications
University of Oldenburg
Oldenburg, Germany
jorge.marx.gomez@uni-oldenburg.de

***Abstract*—Tail biting is one of the biggest problems in pig livestock farming. One indicator that can be observed before an outbreak is the change in tail posture. Studies have shown that days before a tail biting outbreak, a steady increase in hanging tail postures can be observed. A continuous monitoring of this indicator could, therefore, be used to inform farmers of potential problems arising within respective pens. This paper presents a first step in the development of automated monitoring systems for early detection of tail biting indicators by evaluating different approaches for tail posture detection using image data and Deep Learning. Using a dataset consisting of 1000 annotated images, different YOLOv5 object detection models were trained to detect upright and hanging tail postures. The results show that there are significant differences in performance for the detection of upright and hanging class. To further investigate the problem, an EfficientNetv2 image classification model was trained to examine if similar performance differences for the two classes could be observed. Considered in isolation, these differences could be mitigated. However, potentials could not be utilized, as the results of the comparison of the one-step detection of tail posture using YOLOv5 and the introduced two-step detection using YOLOv5 for tail detection and EfficientNetv2 for tail posture classification shows. Based on the discussion of the possible explanations for the inferior performance as well as the summary of the key findings of this paper, we present approaches that can be used as a basis for future research.**

***Keywords*—precision livestock farming; tail biting; tail posture; deep learning; computer vision.**

I. Introduction

Structures of modern pig livestock farming, and pork production have been undergoing major changes in recent years. Data from the Federal Statistical Office in Germany shows the contrary trend of steadily decreasing numbers of farms [1] with simultaneously increasing numbers of animals per farm [2], which makes individual animal management and welfare monitoring increasingly difficult. Meanwhile, the slaughter price has remained volatile for several years [3], making it also more challenging for farmers to maintain pig livestock farming economically viable. At the same time, politics and society alike are calling for more sustainable and more animal-friendly husbandry [4], putting additional pressure on the farmer. 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 utilize modern camera and sensor technologies to enable automatic real-time monitoring in livestock production to supervise animal health, welfare, and behavior [4] [5]. This involves the automated acquisition, processing, analysis, and evaluation of sensor-based data like temperature, ammonia, or CO_2 concentration [6] as well as video data [7] [8]. The combination of these different types of information and data sources hold the potential to enable data-driven assistance systems that support farmers in their daily work, creating more time again for more animal- and welfare-oriented husbandry.

One of the biggest problems in conventional pig livestock farming is tail biting [9]. Tail biting describes a behavioral disorder in pigs and is defined as the intentional damaging of the tail of one pig by another pig, which can result in injuries of varying severity [10]. Not only can this have significant consequences for the health and welfare of the individual pig, but the farmer can also suffer economic damage as a result. Tail biting can negatively impact the growth of affected pigs, which, in addition to incurring additional veterinary costs and labor, also has adverse economic consequences for the farmer [11]. Tail biting is a multifactorial problem that potentially results from a number of different internal as well as external factors and can be separated into two phases: the pre-injury phase and the post-injury phase [12]. To detect tail biting before the actual outbreak, the indicators of the pre-injury phase are of particular importance. In their literature review, Schukat and Heise aggregated animal-specific indicators observed in different studies prior to the onset of tail biting [9]. They concluded that especially the change in tail postures was a consistent indicator that could be observed before an outbreak of tail biting. In each of the investigated studies, it was examined that the number of stuck tails increased steadily in the course leading up to the outbreak and that the ratio of curled and lowered or stuck tails shifted strongly [9].

Current developments in Deep Learning (DL) and Computer Vision (CV) could provide the tools to potentially detect and analyze these indicators automatically using video data. Video data is already being used as a basis for a variety of comparable use cases in pig PLF, many of which can be found in literature. Considering practical applications such as the counting of pigs [13], the tracking of pigs over specific time intervals [14], the detection of aggressive behavior among group housed pigs [7], or the automatic weight estimation [15], there are a number of use cases which are addressed by utilizing image data based on camera recordings. For this reason, this paper will present a method for automated recognition of tail posture based on video data.

The paper is structured as follows: In Section II, the current state of the art in the field of tail posture detection and classification is presented. The primary focus lies on papers that apply DL models and architectures, their respective performance as well as the general state of research regarding the observation of pre-injury indicators aside from the DL and CV domain. In Section III, materials and methods are presented. This section covers the data acquisition, model selection, definition of classes as well as label strategy, dataset description and creation as well as the description of the general test environment and setup. In Section IV, the results are presented based on quantitative evaluation metrics, applying standard evaluation metrics for bounding box prediction. The results of the trained tail posture detection models are also examined and evaluated in this section. Section V offers a discussion of the obtained results and Section VI summarizes the key findings of this paper and presents an outlook on how further research can be conducted on the topic of tail posture classification in the future.

II. Related Work

Tail biting is a subject that has been extensively researched in the literature. Already in 1969, van Putten investigated tail biting among fattening pigs and concluded, that tail biting is induced by various factors and hence describes a multi-factorial problem [16]. Since then, other studies have investigated the issue in greater depth. In 2001, Schrøder-Petersen and Simonsen summarized the research published on the issue of tail biting up to that time. It gives an additional overview of both internal and external risk factors that could induce tail biting [12]. In a similar study, Moinard, Mendl, Nicol and Green also investigated different risk factors that can increase the likelihood of tail biting as well as factors that can potentially reduce it [17]. However, tail posture as a potential indicator for early detection of tail biting is not mentioned in any of the previous referenced studies. Schukat and Heise provide the most recent overview of indicators that can be observed prior to the onset of tail biting [9]. In addition to a general increase in activity inside the pen, an increase in various behaviors such as chewing or other hostile interactions and other specific behaviors such as the tail-in-mouth behavior, the change in tail posture prior to the onset of tail biting was particularly observed in the examined studies.

Although many different use cases have already been investigated in the context of pig PLF using methods from the field of DL and CV, the issue of tail biting or the tail detection in general appears to be a vastly underrepresented subject compared to other topics in that domain. In our literature search, we were only able to identify six papers that investigated the prediction of tail biting or other related topics such as tail detection or tail posture classification based on methods from the field of machine learning (ML), DL and CV. The query that was used to search the scorpus literature database can be found in Table I. During review of the obtained papers based on the literature search, one paper was discarded as it only presented a concept for developing an early warning system for tail biting and did not yet present results regarding the performance of different models in detecting tails and their posture [18]. The remaining five papers could be categorized into two groups depending on the type of data that has been used in the respective use case: *sensor-based* and *image-based* approaches. In the following, the results of the papers of the respective category are presented.

TABLE I
Search Query

TITLE-ABS-KEY((”pig” OR ”piglet”) AND (”tail biting” OR ”tail detection” OR ”tail posture”) AND (”deep learning” OR ”computer vision” OR ”machine learning” OR ”machine vision”))

A. Sensor-based use cases

Domun and Pedersen used sensor data like water consumption of individual pigs, pen level temperatures and different indoor climate data like ventilation, cooling, heating, and humidity to train an algorithm based on a bidirectional Long Short-Term Memory (LSTM) and feedforward neural network architecture to predict tail biting with an Area under the Curve (AUC) of 0.782 [19]. Larsen and Pedersen adopted a similar approach and achieved comparable results by using water consumption of individual pigs and temperature data at pen level to train an Artificial Neural Network (ANN) to predict the outbreak of tail biting with an AUC of 0.75, while producing false alarms in 30% of the days [20]. The authors concluded that future research should focus on more event-specific predictors like the tail posture and the development of systems to monitor these indicators, which we adopt in the scope of this paper.

B. Camera-based use cases

Two different approaches for data collection and data usage can be found in this category. In [5], 3D cameras in combination with Linear Mixed models were used to classify tail posture of individual pigs. The cameras were located above the feeder and pointed vertically down, covering one third of the pen. Validated against human observers, the proposed algorithm did best in the detection of tucked tails with an accuracy of 88.4%. However, the detection accuracy for curly tails, which was the most commonly observed class in the dataset, resulted in a score of 41.7%, which leaves room for improvement. It should also be noted that 3D camera systems are much more expensive than conventional 2D cameras, which also makes it difficult to transfer these systems into agricultural practice. Ocepek et al. [21] present the only research that is comparable to the approach followed in this paper. Using 2D cameras mounted under the roof for data collection, they applied a YOLOv4 object detection model to detect *straight* and *curled* tail postures with an Average Precision (AP) of 90%. However, the model was only trained on 30 images and other important metrics for performance evaluation of object detection models such as Precision (P), Recall (R) and mean Average Precision (mAP) for different Intersection over Union (IoU) thresholds are missing, which makes the results not representative in our opinion.

Based on the identified and analyzed literature, clear research gaps can be identified in the area of tail detection, tail posture detection, and general use cases in the field of tail biting, especially when using image data. Therefore, this paper will address the following research gaps:

a) Usage of a larger, more representative data set consisting of 1000 manually annotated images.
b) Provision of relevant metrics that allow for a better evaluation of model performance.
c) Evaluation of differently complex model architectures in terms of size and number of parameters and their influence on performance.

III. Materials and methods

A. Data acquisition

Data collection was conducted within the DigiSchwein project at the agricultural research farm for pig husbandry of the Chamber of Agriculture Lower Saxony in Wehnen. Within the project, video recordings from both piglet rearing and fattening were collected and stored for analysis. An AXIS M3 16-live network camera was used for video recording in the piglet rearing pens, while the VIVOTEK IB9367-H model was used for the fattening pens. In both cases, the cameras were mounted beneath the ceiling to capture the entire pen from a top-down view. Since piglets in piglet rearing are much more active, move much faster compared to pigs in fattening pens and are also a lot smaller, recording within piglet rearing was conducted with 30 Frames Per Second (FPS) and a resolution of 2688×15120, while in fattening pens recording was done with 10 FPS and a resolution of 1920×1080. The higher resolution enabled us to capture the more rapid and spontaneous movements of the piglets, and to provide the necessary level of detail to ensure that the tails were always clearly visible in the images. Since the pigs in the fattening are much larger and slower, we lowered resolution as well as the number of FPS to reduce the amount of memory needed to store the videos. Figure 1 shows some example images that were extracted from the video recordings.

B. Model selection

The model selection for the task of tail posture detection was conducted based on defined selection criteria. These criteria were derived based on models and architectures that were already used in pig PLF literature as well as on the requirements for PLF systems that have been mentioned in the PLF literature. The following criteria were defined:

1) **Prediction accuracy**: The prediction of the respective models should be as accurate as possible [22].
2) **Prediction speed**: Model inference should be in real-time [23].

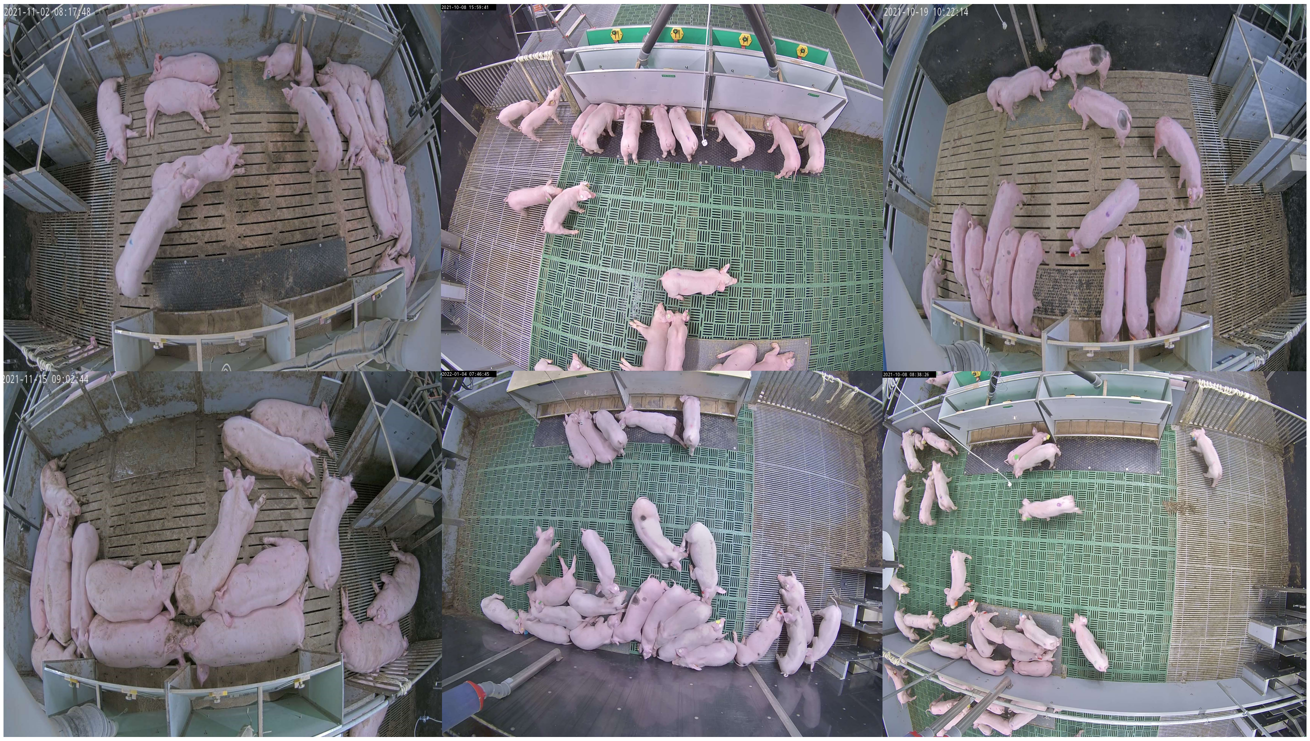

Fig. 1. Example images.

3) **Cost efficiency**: The respective models should be as resource efficient as possible to allow a potential deployment to low-cost hardware [24].

Since the YOLO architecture has already been used in the PLF literature, matches the established criteria by balancing performance, speed, and resource requirements, and has an extensively documented repositories as well as an active community, YOLO was chosen as a baseline architecture for further analysis. YOLOv5 is the latest instalment of the YOLO architecture, but there is currently no official paper for this release. The latest paper release is related to YOLOv4 [25]. YOLOv4 applies specific methods and concepts summarized under the terms bag of freebies and bag of specials to improve accuracy and execution speed compared to YOLOv3 and other architectures such as EfficientDet. The performance was further improved by introducing a network scaling approach by modifying model depth, width, resolution, and structure [26]. The comparison of the two official implementations of YOLOv4 and YOLOv5 resulted in the selection of the YOLOv5 implementation, as it was more suitable for the context of this paper.

C. Label strategy

Before we started to create the dataset, we defined a label strategy in which we specified class selection, image selection and other aspects regarding dataset annotation. To select the label classes, we first identified and compared the different tail postures previously mentioned in the literature and subsequently validated them by an expert group consisting of farmers, veterinarians, and researchers within the DigiSchwein project. Schukat and Heise [9] mention the *curled*, *hanging*, and *stuck* tail posture [9]. D'Eath et al. [5] defined similar tail postures, with the difference that instead of *hanging* they specified the class *loose*, which was further separated into the classes *low loose* and *high loose*. Ocepek et al. [21] distinguish tail posture into *straight* and *curled*, with the *straight* class including tucked tails as well. Furthermore, it is described that both classes were only annotated if the respective pig in the image was standing. If it was lying, the respective pig was not annotated. After a discussion in the expert group of the DigiPig project, it was initially decided to separate the posture classes into three classes similar to [9] and [5]: *upright*, *hanging* and *stuck*. Figure 2 shows examples for each defined class. However, unlike [21], we decided to annotate every visible tail object and its respective posture on the images, regardless of whether it is lying or standing. Reason for this is the assumption that the distinction of the pigs' posture and hence the decision whether to annotate an object or not is not being represented in the training dataset. Thus, there is a possibility that the algorithm may also not learn these relationships. Furthermore, if the tail objects are labelled inconsistently without a reason for the distinction being represented in the dataset, it could negatively impact the performance of the model. The raw images used for dataset creation were extracted from the video recordings that were captured inside the DigiSchwein project. For video and frame selection, we specified the following guidelines:

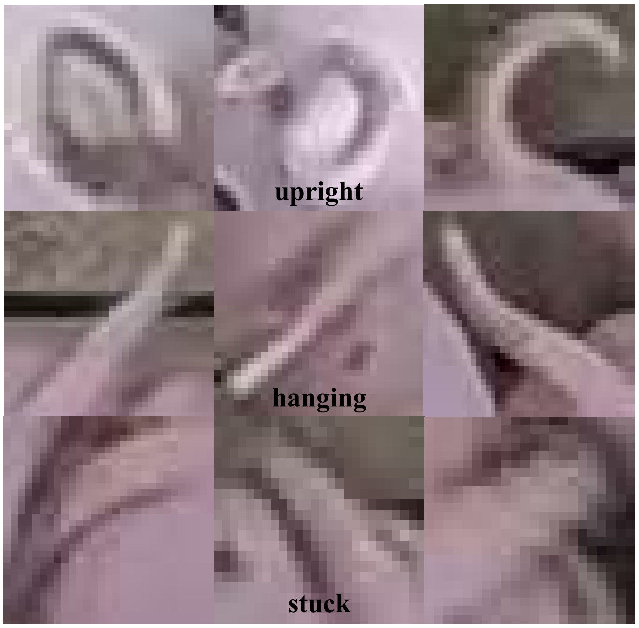

Fig. 2. Tail posture examples.

1) Inclusion of images from piglet rearing and fattening as well as different camera angles, backgrounds, and perspectives to ensure as much data heterogeneity as possible.
2) Inclusion of images with different activity levels within the respective pens to ensure a balanced ratio of standing and lying pigs and a balanced distribution of pig positions and locations inside the pen to further increase data heterogeneity.
3) Balancing the number of each defined tail posture so that none of the defined classes are over- or underrepresented.

Especially the last point caused problems during the data collection and annotation process. The class *stuck* was extremely underrepresented in the extracted images and it was difficult to find additional video recordings in which the respective class could be identified. Additionally, initial results based on a prototypically trained model showed that the class *stuck* was difficult to detect because of this class imbalance. Based on a dataset containing 400 images, a YOLOv5m model trained with an image size of 1280×1280 achieved a mAP0.5:0.95 of 0.277, which also improved just slightly as the number of images increased. To deal with this imbalance, we decided to merge the *stuck* class into the *hanging* class, creating a better balance between the defined classes.

Another challenge that emerged during the annotation process was the annotation of the tails of lying pigs. In some cases, it could not be clearly determined to which class the respective tail could be assigned to. Figure 3 shows some examples for these cases. Even after discussing the issue within the expert group of the DigiSchwein project, no

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-994-2

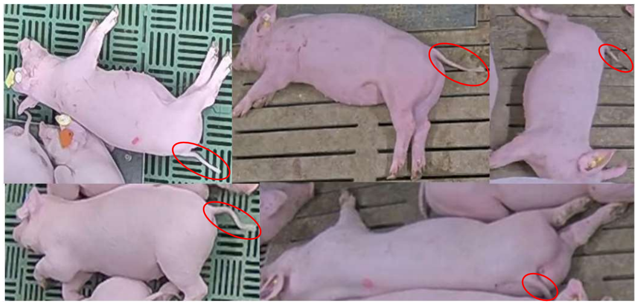

Fig. 3. Difficult to distinguish classes.

clear consensus could be reached. In essence, two separate approaches have emerged when explaining the determination of the class: One group argued that lying pigs, as the ones shown in Figure 3, must have a hanging tail posture because, when in a relaxed or almost relaxed state, the tail would have a hanging position due to gravity alone. The other group argues based on the orientation of the tail of the lying pigs and tends to classify these tail postures as upright. It was difficult to decide how to translate the results of this discussion into the label strategy, as both ways of reasoning are comprehensible. Ultimately, it was decided to assign these instances to the *hanging* class.

D. Dataset description

A dataset consisting of 1000 images with a total of 12802 high quality bounding box annotations was created. By following the data extraction and label strategy, a proper class balance of 6391 *upright* and 6412 *hanging* annotations was ensured. The open-source tool *Labelme* was used to annotate the images for model training and evaluation [27]. Images were extracted from video recordings that were collected in the DigiSchwein project and subsequently annotated according to defined label strategy described in Section III-C. To reduce the overall annotation time, the dataset creation process was divided into different phases:

1) A sample of 200 out of the total 1000 images were manually annotated. This sample was first randomly extracted from the overall data pool and then inspected to ensure data variety was sufficient in the sample
2) A pretrained YOLOv5 model based on the YOLOv5m checkpoint was trained using the annotated sample. The model was subsequently applied on the unlabeled data to generate predictions.
3) The predictions of the model were transformed into the Labelme JSON format and loaded into the annotation tool, where the predictions were subsequently checked by a human annotator. If the given annotations for the respective image were inaccurate or incorrect, they were adjusted manually within the Labelme tool. After the 200 images were reviewed, the model was re-trained with the additional data and the process re-started from step 1) until all 1000 images were annotated. This ultimately reduced the amount of manual label work.

E. Test environment

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. For the object detection task of detecting tail postures of pigs, the YOLOv5 implementation of Jocher et al. [28] was applied. Standard parameters were used for training of each selected model variant. Data augmentation is applied in form of image mosaic and mix-up, random image flip, image rotation, image scaling as well as hue saturation value augmentation. Each model was trained for a maximum of 300 epochs with a batch size of 64 for the 640×640 models and 16 for the 1280×1280 image size models. The YOLOv5 model checkpoints *s*, *m* and *l* as well as the updated versions *s6*, *m6* and *l6* of the respective variants were selected for training and evaluation. Training was stopped early if performance did not improve over several epochs or if validation losses was increasing over several epochs to avoid overfitting. For each trained model, the epoch with the best result on the validation set was saved and used for evaluation. We used an 80-20 split for train and test data.

IV. Results

A. Evaluation metrics

The commonly used metric for evaluation and benchmarking object detection models is the mAP, which is the mean of the AP averaged over all defined classes based on a set of different IoU thresholds [29]. IoU is defined as the similarity between the ground truth annotation and the predicted annotation present in the image and is determined by dividing the intersection with the area of union [30]. Common thresholds to calculate the mAP are values in the range 0.5 to 0.95 with a threshold step size of 0.05, represented in this paper as mAP0.5:0.95 [31]. The higher the IoU threshold, the less margin is allowed in the deviation of the ground truth bounding box and predicted bounding box to be labelled as correct, so the AP is usually lower at higher thresholds. P and R for each tested model variant are also provided, describing what portion of the positive predictions were correct and what portion of the positive predictions was detected correctly, respectively. We also included inference time as well as the number of parameters for each tested model variant. The number of parameters describes the model size and can affect the required hardware to train and operationalize the respective model, having direct effect on the inference time.

B. Quantitative results

The results are shown in Table II, while Table III shows the inference time of each evaluated model variant as well as their number of parameters. The old and updated model variants showed a similar inference time within the experiments when using the 1280 image size, which is why they are merged with the updated variants. During testing, different inference times were observed when performing the same evaluation run multiple times. The table, therefore, shows the respective

TABLE II
RESULTS TAIL POSTURE DETECTION.

		Tail Posture Detection														
		640				*1280*									
Class	*M*	*P*	*R*	*mAP0.5*	*mAP:0.95*	*M*	*P*	*R*	*mAP0.5*	*mAP:0.95*	*M*	*P*	*R*	*mAP0.5*	*mAP:0.95*
all	**s**	0.802	0.675	0.716	0.301	**s6**	0.875	0.813	0.857	0.459	**s**	0.865	0.813	0.852	0.460
upright		0.849	0.778	0.822	0.356		0.923	0.862	0.911	0.505		0.906	0.853	0.802	0.511
hanging		0.755	0.572	0.610	0.246		0.827	0.763	0.803	0.412		0.824	0.772	0.802	0.410
all	**m**	0.835	0.755	0.801	0.394	**m6**	0.861	0.840	0.872	0.486	**m**	0.883	0.838	0.867	0.491
upright		0.907	0.827	0.884	0.445		0.901	0.883	0.919	0.526		0.922	0.887	0.915	0.540
hanging		0.763	0.684	0.718	0.342		0.822	0.797	0.825	0.446		0.845	0.789	0.819	0.443
all	**l**	0.850	0.720	0.790	0.366	**l6**	0.871	0.828	0.857	0.487	**l**	0.885	0.844	0.879	0.511
upright		0.897	0.816	0.879	0.424		0.906	0.876	0.903	0.531		0.919	0.885	0.922	0.552
hanging		0.803	0.624	0.700	0.309		0.836	0.780	0.811	0.444		0.85	0.803	0.836	0.469

TABLE III
NUMBER OF PARAMETERS AND INFERENCE TIME.

Model	***Parameter (m)***	***Inference (ms)***	
s	7.2	640	1.2
m	21.2		2.9
l	46.5		4.4
s6	12.6	1280	5.2
m6	35.7		9.3
l6	76.8		17.9

mean of the observed inference times. Each trained model was evaluated three times after we noticed the problem in order to investigate the deviations in the inference times in more detail. However, an exact cause could not be determined. Overall, it can be observed that, with one exception, the performance of tail posture detection can be increased when a greater image size and a larger model variation is used. The training of the YOLOv5l model with an image size of 640 had to be stopped early because performance did not improve after several epochs as well as overfitting that could be observed after around 100 epochs, which subsequently resulted in an overall reduction of the measured performance compared to the YOLOv5m variant. This could be due to the fact that the YOLOv5l model is too complex and too big for the considered use case, which is why this model and image size combination seems unsuitable. In general, it can be observed that the performance results for the *s*, *m* and *l* model variants trained on a 640 image size are insufficient. With an mAP0.5:0.95 of 0.394 a P and R of 0.835 and 0.755 over all classes, the YOLOv5m achieves the best results in this image size category. Although this combination has the fastest inference time with 2.9 ms per frame as well as the lowest number of parameters and, therefore, has the lowest hardware requirements, the results are not sufficient enough to further investigate this combination of model and image size. However, an improvement in all measured metrics can be observed when using the 1280 image size for the *s*, *m* and *l* model variants. Compared to the YOLOv5s 640 combination, a mAP0.5:0.95 of 0.46 can be achieved when increasing the image size to a 1280 × 1280 resolution, resulting in an overall increase in performance of almost 52% in terms of mAP0.5:0.95. This also significantly increases the number if parameters of the model as well as the inference time, which increases to 9.3 ms in comparison to the 2.9 ms. Despite this, real time inference can still be ensured when using appropriate hardware, which is why the YOLOV5m with an image size of 1280 × 1280 offers the best balance between accuracy, speed, and hardware requirements out of the tested model variants. When comparing the performances of the *s*, *m* and *l* versions with the updated *s6*, *m6* and *l6* variants, it is noticeable that the updated models are, with the exception of minimal improvements in P and partial R in some cases, consistently lower than the measured performances of the older variants. For this reason, the updated variants are not further considered in the quantitative analyzes of the results. Overfitting, which was previously observed for the YOLOv5l 640 variant, does not occur when using the larger image resolution. In fact, using the larger model variant improved performance in all measured metrics, but the difference is much smaller than when changing from the YOLOv5s to the YOLOv5m variant. This might mean that in terms of parameter number and model complexity, a bottleneck has been reached and further performance improvements cannot be achieved by just using more complex models, but rather by providing better training data, more training data, or different approaches for tail posture detection in general. This becomes even more evident when comparing the performance of the *upright* and *hanging* class. For each model variation evaluated, P, R, mAP, and mAP0.5:0.95 are significantly lower for the *hanging* class compared to the *upright* class, although both classes are relatively balanced in the train and test set. This may be due to the higher complexity of the *hanging* class caused by the merging of the *hanging* and *stuck* class, previously discussed in Section III-C as part of the label strategy.

To further investigate this problem, the method presented in [32] was adapted by separating the tail posture detection into two separate stages: A detection step, where only the *tail* class is detected and an subsequent image classification step, where the detected bounding boxes of the *tail* detection model are classified into the defined posture classes *upright* and *hanging*. The idea behind this is that by further merging the classes *upright* and *hanging* into the class *tail*, the highest possible level of representation could be obtained, so that the classes would no longer be considered disjoint from each other and thus performance in tail detection could potentially increase. The actual classification of the posture will then be moved to an image classification model, which will eventually also be used to investigate whether the same differences in performance can be observed for the *upright* and *hanging* classes as with the tested YOLOv5 model variants. In the final step, the two presented methods of one-step tails posture detection and two-step tail posture detection consisting of an object detection and image classification model will be compared and benchmarked based on their performance. The results are presented in the following sections.

C. Results image classification

We first wanted to determine whether image classification models have similar problems in classifying the *hanging* class. Based on the selection criteria defined in Section III-B as well as in [32], we selected the EfficientNetV2-B0 model for training. Based on the annotated object detection dataset described in Section III-D, we created an image classification dataset by extracting the bounding boxes from the object detection dataset using the annotated coordinates and subsequently saved them as separate files. For model training, the official Keras implementation of EfficientNetV0 has been used. Transfer learning was applied by first freezing all but the top layers when initializing the model and utilizing pre-trained ImageNet weights. Second, the layers were unfrozen to fit the model on the new data. In both steps, the model was trained for 30 epochs with a batch size of 128 and an input size of 224×224. Data augmentation was applied in form of random horizontal flipping, random zoom, random rotation as well as random crops and random contrast changes. Training was stopped early when accuracy and validation accuracy intersected, and accuracy continued to increase while validation accuracy stagnated or decreased to avoid overfitting. Sigmoid activation function was used in the last layer while Adam was used as optimizer. Binary cross entropy was specified as the loss function. The results are shown in Table V. Performance in terms if *P*, *R* and *F1* are almost identical for both classes. With a *P* of 0.970, the *hanging* class even achieves a slightly higher performance than the *upright* class, while *R* is still slightly lower for the *hanging* class. Unlike the YOLOv5 model, the problem of distinguishing and classifying the *hanging* class do not seem to be present here, but to ultimately verify whether the approach of separating tail posture detection into an object detection and image classification step can improve performance, the entire process must be considered. Therefore, the following presents the results of the object detection model, which only detects the merged *tail* class and that serves as a preceding step before the image classification step.

D. Results tail detection

To create the data set for detecting the higher-level class *tail*, the *upright* and *hanging* labels of the object detection data set were merged and overwritten in the label files. To ensure a direct comparison, the same combinations of YOLOv5 model variants and image sizes were applied as in Section II for model training and evaluation. The results, presented in Table IV, show that merging the two classes can improve the performance of every measured metric. Comparing the two YOLOv5m 1280 variants, a mAP0.5:0.95 of 0.530 can be achieved compared to the mAP0.05:0.95 of 0.491, resulting in a significant increase in performance. P and R can also be improved, while R still remains lower than P. Up to this point, it can be concluded that separating the tail posture detection into an object detection and an image classification step can, in isolation, lead to performance improvements in the respective tasks. However, the results are only beneficial if the combination of the *tail* object detection model and the image classification model translate into a performance improvement that surpasses the results of the model presented in SectionII. This will be examined in the following section.

E. Comparison

In order to verify whether the presented approach can lead to performance improvements, we compared the performance

TABLE V
RESULTS TAIL CLASSIFICATION.

		Tail Classification		
		224		
Class	*Model*	*Precision*	*Recall*	*F1*
all	B0	0.965	0.970	0.970
upright		0.960	0.980	0.970
hanging		0.970	0.960	0.970

TABLE IV
RESULTS TAIL DETECTION

		Tail Detection													
		640				*1280*									
Class	*M*	*P*	R	mAP0.5	mAP0:0.95	M	P	R	mAP0.5	mAP:0.95	M	P	R	mAP0.5	mAP:0.95
tail	**s**	0.879	0.778	0.831	0.381	**s6**	0.905	0.848	0.898	0.474	**s**	0.919	0.861	0.905	0.493
tail	**m**	0.879	0.813	0.851	0.418	**m6**	0.899	0.880	0.916	0.522	**m**	0.926	0.877	0.918	0.530
tail	**l**	0.888	0.805	0.857	0.422	**l6**	0.917	0.881	0.921	0.531	**l**	0.921	0.884	0.924	0.547

of the one-step YOLOv5m and YOLOv5l tail posture detection trained on a 1280×1280 image size with the two-step approach consisting of the YOLOv5m and YOLOv5l model for *tail* detection and the EfficientNetV2-B0 model for subsequent image classification of the detected bounding boxes into *upright* and *hanging*. Since the YOLOv5m variant offers the best balance of performance, speed, and hardware requirements and the YOLOv5l variant gives the best overall performance when disregarding speed as well as hardware requirements, these model variations were selected for comparison. We use a customized version of the *val.py* script of the YOLOv5 repository, which we adapted to integrate the *tail* detection model as well as the EfficientNetV2-B0 model for image classification in the evaluation pipeline.

The comparison of both approaches, presented in Table VI, shows that the combination of the object detection and image classification methods cannot improve the performance for tail posture detection. The opposite is true, as the direct comparison of the results of the YOLOv5m model variants reveals a decrease in performance for all measured metrics. The difference in performance between the *upright* and *hanging* class is also still present, it even increased in direct comparison. The same observations apply when comparing the results for the YOLOv5l model variations. However, it can also be observed that, as the accuracy of the predicted bounding boxes of the *tail* object detection model used in the pipeline increases, the overall classification of tail posture can also be increased. Although the difference between the map0.5:0.95 of 0.530 and 0.547 for *tail* detection with YOLOv5m and YOLOv5l respectively is not significant, it can lead to a similar performance increase for the two-step tail posture detection based on the combination of object detection and image classification. This results in almost identical performance of the YOLOv5l model variant in combination with the EfficientNetv2-B0 compared to the YOLOv5m variant for one-step tail posture classification.

V. Discussion

Given that, in isolation, both the merging of the *upright* and *hanging* class into the *tail* class led to a more accurate detection and that the subsequent image classification for tail posture classification into *upright* and *hanging* was able to avert the problems regarding the detecting of the *hanging* class presented in Section IV-B, it was surprising that the combination of the two approaches achieved inferior results in comparison. However, a closer look at the data as well as the results reveals a possible explanation, which will be discussed in the following. The results of Section IV-E already demonstrated that the accuracy of the object detection and image classification pipeline is dependent on the accuracy of the object detection model. The more accurate the object detection is, the more accurate the final classification by the image classification model will be. Possible explanation for the inferior performance is that the image classification model was trained using perfectly cropped image data for the *upright* and *hanging* classes, which cannot be provided in that form in inference mode due to the currently existing inaccuracy of the *tail* detection. The image crops provided by the *tail* detection, on the basis of which the posture classification model is supposed to categorize the tail posture, are in terms of the mAPO.5:0.95 of 0.530 and 0.547 for the YOLOv5m model and the YOLOv5l model insufficiently accurate, which is why the image classification model is provided with input images that may not fully capture the targeted *tail* object. Thus, the provided input data by the object detection model may deviate from the actual training data, in which the targeted *tail* object could be represented under ideal conditions. This discrepancy in the data may lead to inferior results when comparing the two approaches. However, this will need to be further validated in future research.

VI. Conclusion

In summary, the following findings can be derived from the results obtained in this paper. Table IV as well as the examination of the results in Section IV-B each show, that performance for one-step tail posture classification based on the YOLOv5 object detection architecture can be increased when larger models and larger image sizes are used for training. However, this performance increase does not scale infinitely, but seems to decrease as the number of model parameters increases. Thus, performance cannot simply be increased by using larger models and larger image sizes. It was also observed that there are large performance differences between the *upright* and *hanging* class. The results in Section IV-C show that, in isolation, the observed differences in performance can be mitigated by using image classification

TABLE VI
Comparison of approaches.

		Tail Posture Detection								
		Object Detection				***Object Detection + Image Classification***				
Class	*M*	*P*	*R*	*mAP0.5*	*mAP:0.95*	*M*	*P*	*R*	*mAP0.5*	*mAP:0.95*
all	**m**	0.883	0.838	0.867	0.491	**m + B0**	0.851	0.803	0.808	0.473
upright		0.922	0.887	0.915	0.540		0.853	0.886	0.862	0.522
hanging		0.845	0.789	0.819	0.443		0.849	0.720	0.754	0.424
all	**l**	0.885	0.844	0.879	0.511	**l + B0**	0.847	0.807	0.823	0.492
upright		0.919	0.885	0.922	0.552		0.846	0.892	0.869	0.537
hanging		0.850	0.803	0.836	0.469		0.848	0.722	0.778	0.447

for tail posture classification. In general, tail detection, as the results in Section IV-D show, can also be improved by merging the *upright* and *hanging* class into the higher-level class *tail*. However, when considered as a whole, the indicated potentials of the object detection model for tail detection and the image classification model for tail posture classification cannot be utilized, as the comparison of the results of the one-step and two-step tail posture classification approach in Table VI revealed. The *hanging* class is not only a problem in terms of detection, but also in terms of annotation, which is also reflected in the obtained model results in Table II. Especially lying pigs seem to aggravate this problem, since it is not always evident to which class the object under consideration can be assigned to. In general, it is questionable whether the tail posture of lying pigs should be included at all as a relevant indicator for tail posture monitoring or, if included, how to properly handle them.

One approach to deal with the identified problems could be to simply include more training data, more diverse training data, better training data or, if the previous solution approaches indicate that the problem is not data related, to find or select a different approach for tail posture detection. The former could particularly help to improve the performance of the presented two-step tail posture classification approach, since the performance of the two-step approach is positively correlated with the performance of the tail detection model, so that an improvement in accuracy for the tail detection model can lead to a better classification of the cropped bounding boxes. However, it is also possible that the problem in detecting the *hanging* class cannot be solved by simply adding more training data. This is where the latter of the mentioned approaches comes into play. One solution approach could be to exclude lying pigs from the tail posture detection process or treat them separately. However, the exclusion of lying pigs is not a trivial task, as it cannot be achieved by simply not annotating lying pigs and their respective tail postures, since for object detection tasks, every object of a defined target class should be annotated in the training set. Thus, the exclusion must be realized in form of a preceding or subsequent step within the tail posture detection process. In future work, we will investigate the approaches of excluding lying pigs from the tail posture detection process or separate handling them based on a preceding pig posture classification step, where we classify detected pigs into *lying* and *notLying* and examine whether performance improvements can be achieved in that way.

VII. Acknowledgments

The project is supported (was supported) by funds of the Federal Ministry of Food and Agriculture (BMEL) based on a decision of the Parliament of the Federal Republic of Germany. The Federal Office for Agriculture and Food (BLE) provides (provided) coordinating support for digitalisation in agriculture as funding organisation, grant number 28DE109A18.

References

[1] Statisches Bundesamt, “Number of pig farms in Germany from 1950 to 2021,” 2021. https://de.statista.com/statistik/daten/studie/1175101/umfrage/betriebe-in-der-schweinehaltung-deutschland/.

[2] Statisches Bundesamt, “Number of pigs per farm in Germany from 1950 to 2021,” 2021. https://de.statista.com/statistik/daten/studie/1174729/umfrage/anzahl-der-schweine-je-betrieb-in-deutschland/.

[3] Bundesministerium für Ernährung und Landwirtschaft, “Slaughter prices of pigs, cattle and lambs,” 2022. https://www.bmel-statistik.de/preise/preise-fleisch/.

[4] 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.

[5] 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, p. e0194524, 2018.

[6] 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.

[7] 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.

[8] 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.

[9] S. Schukat and H. Heise, “Indicators for early detection of tail biting in pigs - a meta-analysis.,” 2019.

[10] N. R. Taylor, D. C. J. Main, M. Mendl, and S. A. Edwards, “Tail-biting: a new perspective,” *Veterinary journal (London, England : 1997)*, vol. 186, no. 2, pp. 137–147, 2010.

[11] R. B. D’Eath *et al.*, “Changes in tail posture detected by a 3D machine vision system are associated with injury from damaging behaviours and ill health on commercial pig farms,” *PLOS ONE*, vol. 16, no. 10 October, 2021.

[12] D. L. Schrøder-Petersen and H. B. Simonsen, “Tail biting in pigs,” *The Veterinary Journal*, vol. 162, no. 3, pp. 196–210, 2001.

[13] G. Chen, S. Shen, L. Wen, S. Luo, and L. Bo, “Efficient Pig Counting in Crowds with Keypoints Tracking and Spatial-aware Temporal Response Filtering.”

[14] J. Cowton, I. Kyriazakis, and J. Bacardit, “Automated Individual Pig Localisation, Tracking and Behaviour Metric Extraction Using Deep Learning,” *IEEE Access*, vol. 7, pp. 108049–108060, 2019.

[15] 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.

[16] G. van Putten, “An Investigation into Tail-Biting among Fattening Pigs,” *British Veterinary Journal*, vol. 125, no. 10, pp. 511–517, 1969.

[17] C. Moinard, M. Mendl, C. Nicol, and L. Green, “A case control study of on-farm risk factors for tail biting in pigs,” *Applied Animal Behaviour Science*, vol. 81, no. 4, pp. 333–355, 2003.

[18] P. Wißkirchen *et al.*, “Early detection of tail biting among pigs on the basis of deep learning: Development concept of a practical early warning system,” pp. 343–348, Gesellschaft fur Informatik (GI), 2021.

[19] Y. Domun, L. J. Pedersen, D. White, O. Adeyemi, and T. Norton, “Learning patterns from time-series data to discriminate predictions of tail-biting, fouling and diarrhoea in pigs,” *Computers and Electronics in Agriculture*, vol. 163, p. 104878, 2019.

[20] M. L. V. Larsen, L. J. Pedersen, and D. B. Jensen, “Prediction of Tail Biting Events in Finisher Pigs from Automatically Recorded Sensor Data,” *Animals : an open access journal from MDPI*, vol. 9, no. 7, 2019.

[21] M. Ocepek, A. Žnidar, M. Lavrič, D. Škorjanc, and I. L. Andersen, “Digipig: First developments of an automated monitoring system for body, head and tail detection in intensive pig farming,” *Agriculture (Switzerland)*, vol. 12, no. 1, 2022.

[22] 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.

[23] 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.

[24] Banhazi, H. Lehr, J. L. Black, H. Crabtree, and D. Berckmans, "Precision Livestock Farming: An international review of scientific and commercial aspects," *International Journal of Agricultural and Biological Engineering*, vol. 5, no. 3, pp. 1–9, 2012.

[25] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020.

[26] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network," 2020.

[27] K. Wada, "labelme: Image Polygonal Annotation with Python," 2016.

[28] Jocher, Glenn et al., "ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations," 2021.

[29] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[30] M. A. Rahman and Y. Wang, "Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation," in *ISVC*, 2016.

[31] 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.

[32] J.-H. Witte and J. Marx Gómez, "Introducing a new Workflow for Pig Posture Classification based on a combination of YOLO and EfficientNet," in *Proceedings of the 55th Hawaii International Conference on System Sciences* (T. Bui, ed.), Proceedings of the Annual Hawaii International Conference on System Sciences, pp. 1135–1144, Hawaii International Conference on System Sciences, 2022.