agriculture

MDPI

Article

Automatic Position Detection and Posture Recognition of Grouped Pigs Based on Deep Learning

Hengyi Ji [1,2], Jionghua Yu [2], Fengdan Lao [2,3], Yanrong Zhuang [1,2], Yanbin Wen [2,4] and Guanghui Teng [1,2,*]

1 College of Water Resources & Civil Engineering, China Agricultural University, Beijing 100083, China
2 Key Laboratory of Agricultural Engineering in Structure and Environment, Ministry of Agriculture and Rural Affairs, Beijing 100083, China
3 Network Center, China Agricultural University, Beijing 100083, China
4 Bureau of Agricultural and Rural Affairs, Datong 037000, China
* Correspondence: futong@cau.edu.cn; Tel.: +86-010-62737583

Abstract: The accurate and rapid detection of objects in videos facilitates the identification of abnormal behaviors in pigs and the introduction of preventive measures to reduce morbidity. In addition, accurate and effective pig detection algorithms provide a basis for pig behavior analysis and management decision-making. Monitoring the posture of pigs can enable the detection of the precursors of pig diseases in a timely manner and identify factors that impact pigs' health, which helps to evaluate their health status and comfort. Excessive sitting represents abnormal behavior when pigs are frustrated in a restricted environment. The present study focuses on the automatic recognition of standing posture and lying posture in grouped pigs, which shows a lack of recognition of sitting posture. The main contributions of this paper are as follows: A human-annotated dataset of standing, lying, and sitting postures captured by 2D cameras during the day and night in a pig barn was established, and a simplified copy, paste, and label smoothing strategy was applied to solve the problem of class imbalance caused by the lack of sitting postures among pigs in the dataset. The improved YOLOX has an average precision with an intersection over union threshold of 0.5 ($AP_{0.5}$) of 99.5% and average precision with an intersection over union threshold of 0.5–0.95 ($AP_{0.5–0.95}$) of 91% in pig position detection; an $AP_{0.5}$ of 90.9% and an $AP_{0.5–0.95}$ of 82.8% in sitting posture recognition; a mean average precision with intersection over union threshold of 0.5 ($mAP_{0.5}$) of 95.7% and a mean average precision with intersection over union threshold of 0.5–0.95 ($mAP_{0.5–0.95}$) of 87.2% in all posture recognition. The method proposed in our study can improve the position detection and posture recognition of grouped pigs effectively, especially for pig sitting posture recognition, and can meet the needs of practical application in pig farms.

Keywords: pig behavior; object detection; deep learning; computer vison

check for updates

Citation: Ji, H.; Yu, J.; Lao, F.; Zhuang, Y.; Wen, Y.; Teng, G. Automatic Position Detection and Posture Recognition of Grouped Pigs Based on Deep Learning. *Agriculture* **2022**, *12*, 1314. https://doi.org/10.3390/agriculture12091314

Academic Editor: Roberto Alves Braga Junior

Received: 13 July 2022
Accepted: 24 August 2022
Published: 26 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Pigs' behavior reflects their health and growth status and affects pork production and economic benefits. The monitoring and recognition of behavior are of great significance for the precision management of pigs [1]. With the development of sensors and video surveillance technology, information and communication technology, big data, and artificial intelligence technology, various sensors are applied to monitor animal behavior. For example, three-axis acceleration sensors are used to monitor the prenatal behavior characteristics of sows in real time [2], pressure sensors are used to monitor the activities of sows during parturition [3], and RFID is used to replace simple ear tags, enabling precision feeding [4]. However, the shortcomings of sensors are gradually being revealed in practical usage. The devices need to be worn externally, which causes stress to the animals. There is also a certain drop rate in the contact and movement of the pigs, and some sensors installed in the field require the intervention of the breeder to take readings [5]. Therefore, non-contact

computer vision technology has been gradually used in the process of pig farming [6–8]. Some researchers have applied computer vision systems to the daily monitoring of pigs, such as for the recognition of drinking behavior, feeding behavior, and fighting behavior [9]. Non-contact computer vision systems are more suitable for the expanding commercial pig farming model.

In recent years, deep learning has been demonstrated to learn higher-level abstract representations of data and extract features from data automatically [10,11]; it has become a central method in the field of computer vision and has achieved multiple successes regarding image classification and object detection. As a further extension of the image classification task, object detection is one of the hotspots in the field of computer vision, not only in the recognition of the classes of an object in an image but also in identifying the area where the object is located and framing it with a bounding box. Object detection based on deep learning is divided into two-stage and one-stage algorithms. Two-stage algorithms first need to generate an anchor box that may contain objects and then perform fine-grained object detection. These algorithms have high accuracy but slow speed and include representative models such as R-CNN [12], Faster R-CNN [13], and SPPNet [14]. Conversely, one-stage algorithms directly extract features in the network to predict the position coordinate of the object's class probability, so they have a better balance in detection speed and accuracy than two-stage models; they are represented by the YOLO [15] series, SSD [16], and CenterNet [17].

The accurate and rapid detection of abnormal behaviors is a prerequisite for taking appropriate and timely measures to reduce their incidence. Detection algorithms are the key basis for pig behavior analysis and management decision-making [18,19]. Seo et al. [20] detected pigs using an NVIDIA Jetson Nano embedded board by reducing the parameters of the 3×3 convolutions in the Tiny-YOLO object detector. Ahn et al. [21] combined the test results of two YOLOv4 models at the bounding-box level to increase the pig detection accuracy from 79.93% to 94.33%. Yan et al. [22] combined feature pyramid attention with Tiny-YOLO to improve pig detection accuracy. Fang et al. [23] improved CenterNet as a pig object detection model by fusing the lightweight network MobileNet and a feature pyramid network (FPN) structure, which reduced the model size and improved the pig detection accuracy. In summary, the YOLO series has been widely used due to its good balance between speed and accuracy, small model size, and easy deployment. However, its detection accuracy needs to be further improved, and there are limitations in detecting target pigs due to low illumination at night and occlusion in crowds, which must be addressed.

The postures of pigs include standing, lying on their sides, sitting, etc. The postures of pigs indicate the growth state and the comfort level of their growing environment. Monitoring their postures can help to detect the precursors of pig diseases quickly, identify factors that threaten the health of pigs in advance, and judge whether or not the pigs are comfortable. Zheng et al. [24] used Faster R-CNN to recognize the postures of lactating sows within a deep learning framework. Zhu et al. [25] used improved dual-stream RGB-D Faster R-CNN to recognize sow postures automatically. Yang et al. [26] developed a CNN-based method to recognize the posture changes of sows in in-depth videos. All the above posture recognition techniques are based on images captured by depth cameras with high financial expenditures. To reduce financial costs, some researchers used 2D cameras to recognize the postures of grouped pigs. Nasirahmadi et al. [27] combined R-FCN and ResNet101 to recognize pig postures such as standing, lying on the side, and lying on the stomach. Riekert et al. [28,29] improved Faster R-CNN to detect the positions of pigs and recognize lying and non-lying postures. Shao et al. [30] extracted individual pigs from pictures of herds through YOLOv5 and then used the DeepLab v3+ segmentation method to extract individual pig contours, followed by recognizing postures in a deep separable convolutional network. Previous research has focused on the recognition of standing and lying postures in pigs, although the sitting posture is equally important for grouped pigs. The normal expression of a sitting posture is considered a maintenance

behavior of pigs, and overuse of the sitting posture is an abnormal behavior, indicating frustration in a restricted environment [31,32]. Notably, the sitting position of pigs is a transitional behavior from lying down to moving. An increase in the use of a sitting position indicates that grouped pigs will change from lying down in a resting state to an active state [33]. Staying in a sitting posture for a long time will increase tactile communication and fighting behavior [34,35]. A pig sits for approximately 100 s per hour on average, and only 1–2 pigs are in the sitting posture simultaneously in a pen containing 12 pigs [33], which demonstrates that sitting is a rare posture. The class imbalance of datasets is one of the most difficult problems to be solved in the automatic recognition of sitting postures. A two-stage algorithm is often used for recognition, with low real-time performance, a large model size, and high hardware requirements; thus, it is not practical for large-scale deployment in pig farms.

At present, the YOLO series is the most well-known one-stage target algorithm because of its small model size, fast speed, and easy deployment. Among the YOLOv1 to YOLOv5 series, YOLOv5 shows the best detection performance. Its lightweight model size ensures its detection accuracy and speed, outperforming the previous best object detection framework, EfficientDet [36]. However, with the development of object detection, academic research has focused on anchor-free, advanced label assignment strategies and end-to-end detectors, which have not yet been applied in YOLOv5. The anchor-based detection head and manual assignment strategy for training are still applied in YOLOv5, indicating that there is still room for improvement in the model's performance. Therefore, Zheng et al. [37] applied improvements such as an anchor-free quality, a decoupling head, and SimOTA to the YOLO series to propose YOLOX. The detection performance of YOLOX on the COCO dataset has exceeded that of YOLOv5 [37]. Since YOLOv5 is still in the process of continuous updating and optimization, its detection performance continues to improve with the version updates. For these reasons, this study combines the new changes in YOLOv5 and YOLOX to improve the detection performance.

To solve the problems mentioned above, we make the following contributions: (1) a human-annotated dataset of standing, lying, and sitting pigs captured by a 2D camera during the day and night in a pig barn was established; (2) a simplified copy, paste, and label smoothing was conducted to solve the problem of class imbalance caused by the lack of sitting postures of pigs in the dataset; (3) an automatic recognition algorithm for pig posture and behavior was realized based on the improved YOLOX.

This paper is divided into four sections. The first presents the research background and significance. The second describes the datasets and processing methods used in detail. The third describes the experimental results and offers a brief discussion of the methods proposed in this paper. The fourth presents the research conclusions.

2. Materials and Methods

2.1. Animals and Barn

The experimental site was in the No. 3 pig barn of the pig nutrition and environmental regulation scientific research base in Rongchang District, Chongqing, China. There were 50 pig pens in the barn; the size of each pen was 4.2 m × 2.5 m, and pens were equipped with one feeding trough and four drinking fountains. The floor type was a semi-slatted floor structure, and the width of the slatted floor was 1.2 m. Data collection was carried out from June to August 2020 and March to April 2022. From June to August in 2020, pigs with an average initial weight of 25 kg were selected for the experiment, and the number of pigs in a pen was 6 or 8. The barn temperature was maintained at 28–30 °C. After the experiment, the average weight of pigs was 45 kg. From March to April in 2022, pigs with an average initial weight of 40 kg were selected for the experiment, and the number of pigs in a pen was 12. The barn temperature was maintained at 18–20 °C. After the experiment, the average weight of pigs was 58 kg. In the pig barn, the feeding trough was manually filled to ensure the free feeding of the pigs at 8:00 a.m. and 2:00 p.m. every day, and the cleaning and disinfection of pens were conducted at 10:00 a.m. and 3:00 p.m. every day.

Pigs were fed a pelleted (corn- and soya-based) commercial diet and had ad libitum access to food and water. During the experiment, a professional veterinarian assessed the health and welfare of pigs through their environment and behavior. No pigs were removed or moved into the study pens during the experiment.

2.2. Dataset

A high-definition 2D camera (Hikvision DS-2CD3326DWD-I network camera, 1920 × 1080P, 30 frames per second, Hangzhou Hikvision Digital Technology Co., Ltd., HaiKang, Shenzhen, China) was installed above the pen and connected to a hard disk video recorder with a 4T hard disk for recording every other day. To capture more pig postures, in this experiment, we selected periods when the pigs were more active, ranging from 8:00 a.m. to 10:00 p.m. in the morning and from 2:00 p.m. to 4:00 p.m. in the afternoon, for recording. Due to the lower activity of pigs at night, the postures of pigs remained unchanged for a long time, so the recording time at night was from 8:00 p.m. to 12:00 p.m. The videos collected at different time periods and in different pens were processed by frame extraction, and the video frame images were extracted every 20 s. With reference to the literature [24–30], we selected three types of pig postures for recognition, namely, standing, lying, and sitting postures. The pig posture descriptions are shown in Table 1. Images of similar, unchanged postures were examined and deleted manually to ensure the accuracy of positions and diversity of postures. Finally, a total of 2743 pictures containing pig postures were collected to form a dataset.

Table 1. Definition and description of pig postures.

Posture	Description
Standing	Upright body position on extended legs, with only hooves in contact with the floor
Lying	The torso was in full contact with the ground, and the legs were bent or extended
Sitting	Partly erected on stretched front legs with caudal end of body contacting the floor

To allow the model to obtain the position and posture information, data annotation was required. In this experiment, LabelImg (https://github.com/tzutalin/labelImg accessed on 29 March 2022), an open-source tool, was used to label images, which were saved as label files in the format required by YOLO. After each image was labeled, a txt file with the same name was generated, which recorded the posture class of the object pig in the image and the center coordinates, height, and width of the marked bounding box. The labeled dataset was divided into a training set, validation set, and test set according to the ratio of 6:2:2. The training set and validation set were used for model training, and the test set was used to test the model with the best results during training. There were 20,105 labels for 2763 pictures after labeling. The number distribution of labels for the dataset is shown in Table 2.

Table 2. Details of data annotation for pig postures.

Posture Classes	Number of Posture Labels			
	Train	Val	Test	Total
Standing	3691	1161	1187	6039
Lying	7639	2624	2539	12,802
Sitting	788	224	252	1264
Total	12,118	4009	3978	20,105

2.3. YOLOv5

Compared with YOLOv1-v4, YOLOv5 improves the performance of the YOLO series. YOLOv5 adopts different width and depth factors to divide the model from small to large

(n, s, m, l, x), while the overall structure of the model has not changed. Due to computing limitations, the version of YOLOv5 used in this paper was YOLOv5s, and subsequent work was based on this version.

YOLOv5 provides a variety of online data augmentation methods. There are the following four methods used in this paper: augmentation HSV, which randomly adjusts the color, saturation, and brightness of the image in the training set; random affine, which randomly transforms the images in the training set into affine transformations; random horizontal flip, which flips the images in the training set randomly; mosaic, which randomly stitches four pictures from the training set into one picture according to the center point. These online data augmentation steps can increase the diversity of the dataset and improve the generalization ability of the models effectively.

The model architecture of YOLOv5s is composed of a Backbone, Neck, and Head (Figure 1). YOLOv5 still follows and optimizes the previous CSP-DrakNet53 in the Backbone. In the first layer of the network, it uses a convolution layer of 6 × 6 instead of the previous Focus module to make it more suitable for existing GPU devices, which is more efficient. Changing all activation functions to the sigmoid weighted liner unit (SiLU) makes calculations smoother. The C3 module containing bottleneck1 is used to replace the previous CSP bottleneck. C3 (Figure 2) is simplified from CSPbottleneck, which has fewer parameters, a faster speed, and lighter weight, and can better fuse features. Optimized FPN and PAN are used in the Neck's main structure, and the CSP structure is also added, comprising a C3 module containing bottleneck2, to increase the feature fusion capability of the model. In addition, the SPPF structure is used to replace the SPP structure. The SPPF (Figure 3) structure passes the input serial through 5 × 5 maximum pooling layers to guarantee that the information obtained is unchanged after fusing the features of different resolutions while accelerating the running speed and reducing the running time. The structure in the Head section is consistent with YOLOv3 and YOLOv4.

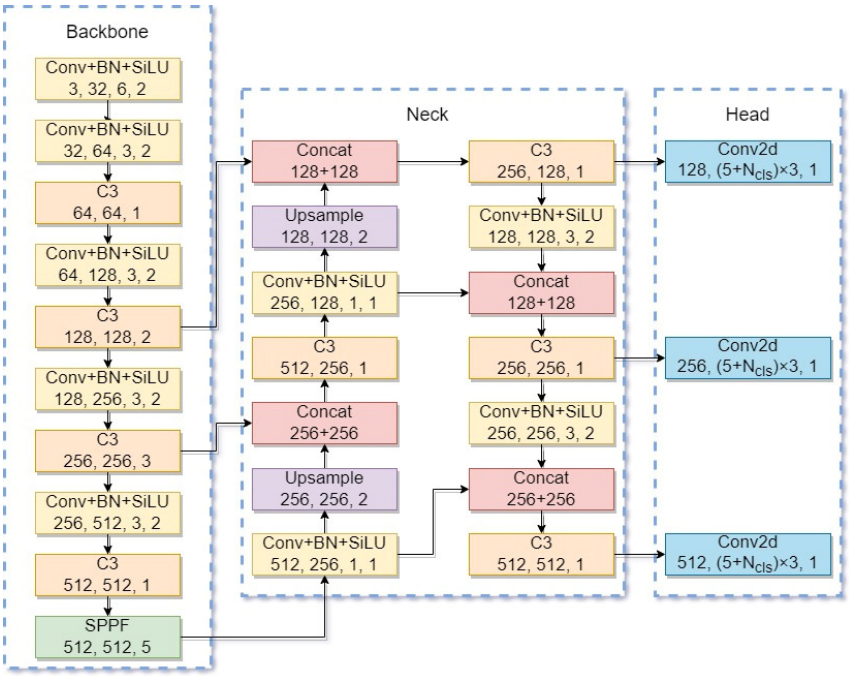


Figure 1. The diagram of YOLOv5s network structure.

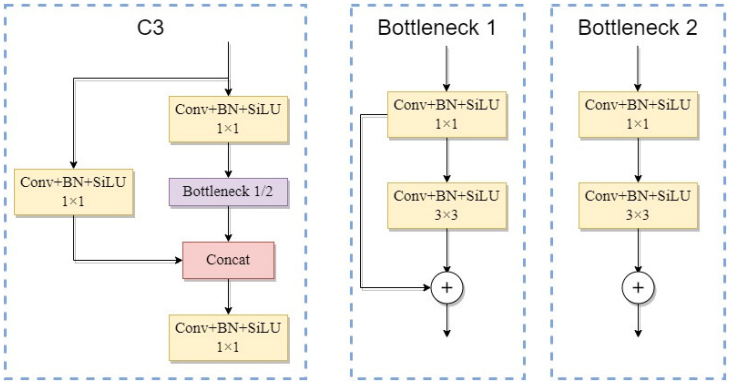


Figure 2. The diagram of C3 structure.

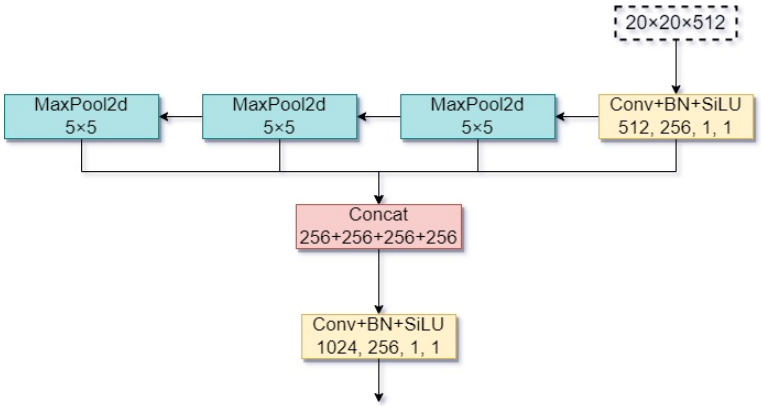


Figure 3. The diagram of SPPF structure in YOLOv5s.

2.4. YOLOX

YOLOX was originally called YOLOX-DarkNet53 based on the improvement of YOLOv3. For ease of comparison with YOLOv5, YOLOX also replaced the Backbone and Neck with the architecture of YOLOv5 and divided the model into s, m, l, and x versions. The version used in this study was YOLOXs. The improvement of YOLOX over YOLOv5 is mainly focused on the Head, which is divided into three parts.

Firstly, the detection head was replaced by the decoupled head. In object detection, the conflict between the classification task and positioning task is a well-known problem. The classification task focuses on which class of the object bounding boxes is closer to the actual class while positioning tasks focus on the location parameters of the object bounding boxes to approach the ground truth box. Using the same feature map to perform the classification task and positioning task leads to feature coupling and task conflict. The decoupled head separates the classification task from the positioning task to avoid feature coupling and task conflict, which improves the model's performance. After decoupling, a total of '$4 + 1 + N_{cls}$' parameters are predicted for each location of the feature graph as follows: '4' indicates the boundary box parameters, '1' is the intersection over union, and 'N_{cls}' means the number of predicted object classes, which is 3 in this study.

Secondly, YOLOv5 was changed from anchor-based to anchor-free. For object detection models that are anchor-based, cluster analysis is needed to determine a set of optimal anchor boxes before training, in order to achieve optimal detection performance. Thus, the anchor boxes obtained by clustering can only be used for specific datasets, which increases the complexity of the detection head and the number of results generated. YOLOX changes from anchor-based to anchor-free by reducing the predicted value of each position from 3 to 1, and directly predicting the X and Y offsets of the object center point relative to the upper left corner of the grid, as well as the object height and width. YOLOX reduces the parameters and GFlops of the model through being anchor-free, making it faster and better in detection.

Finally, we used the SimOTA sample assignment strategy with the anchor-free characteristic; SimOTA has a faster operation time and shorter training time than OTA, avoiding additional optimization parameters, and it has little effect on model accuracy in most datasets.

However, the YOLOv5 used in YOLOX is not the latest version. To take advantage of the new version of YOLOv5 and further increase the detection performance of the model, we combined the Backbone and Neck of YOLOv5s (Figure 1) with the Head of YOLOXs. The new version of YOLOv5s and the model combined with YOLOXs are denoted as YOLOsx in this paper. The YOLOsx network architecture is shown in Figure 4.

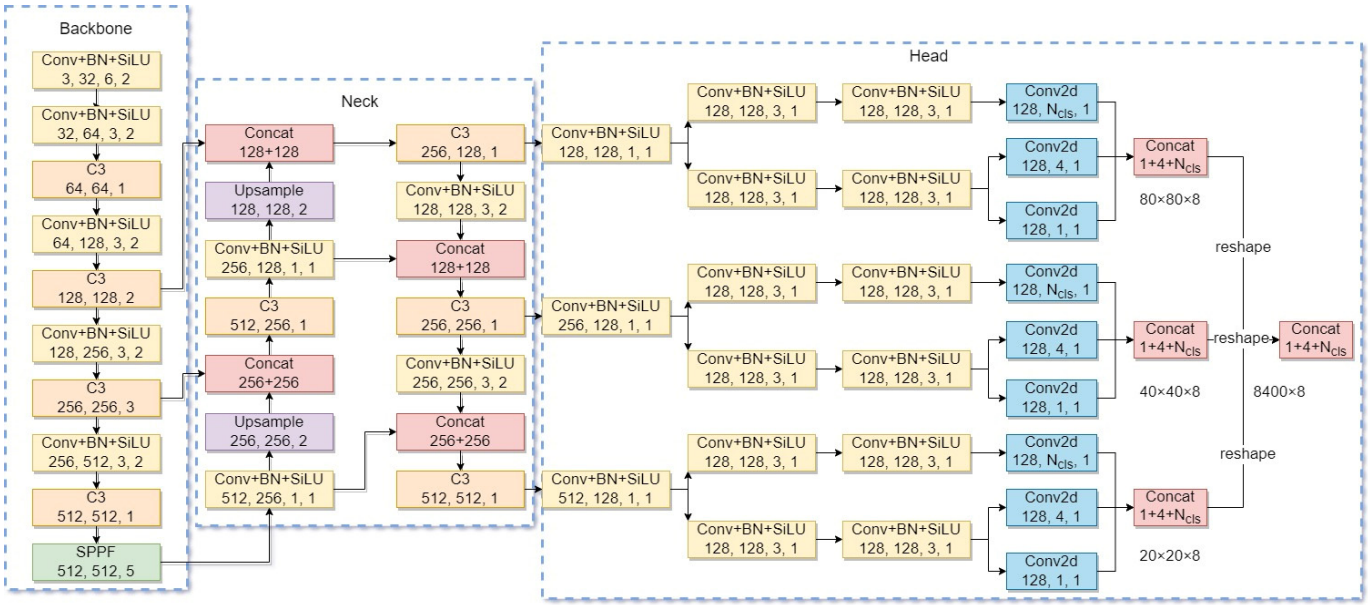


Figure 4. The diagram of YOLOsx network structure.

2.5. Copy-Paste

Since the sitting posture of pigs has always been rare, there were only 788 labels in the training set of pig sitting postures, which was far less than the numbers of other labels. The serious class imbalance problem prevented the model from learning this feature completely. An effective and simplified data augmentation technique, named copy-paste, was used to solve this problem. Meanwhile, offline data augmentation was used to directly increase the number of images containing the sitting class exclusively.

Copy-paste is commonly used for data augmentation in instance segmentation, in which objects with instance segmentation labels are randomly pasted onto an image [38]. Because the object detection model was adopted in this study, the dataset required the addition of instance segmentation labels in order for copy-paste to be used directly. This not only requires a lot of additional labor, but also long periods of time. Therefore, in this experiment, we changed the instance segmentation labels required in copy-paste to the bounding box for object detection. As shown in Figure 5, the simplified copy-paste process filtered the images labeled with the sitting posture in the dataset, then clipped the sitting posture from the image to form a crop dataset according to the bounding box of the sitting posture and pasted the image randomly from the crop dataset into the background image or other images. In this experiment, copy-paste was used for the training set, and the images without pigs in the pen were used as the pasted background images. With this method, we pasted four times randomly on each background image, which means that four additional sitting posture labels appeared in the background image where the copy-paste was completed. To prevent the overlapping of pigs between the crop images of the sitting posture from affecting the performance of the model, the overlap threshold was set to the intersection ratio of the two pigs' bounding boxes equal to 0. In the training set, 600 images after copy-paste were added, without adding the validation set and test set. Therefore, the

number of sitting class labels in the training set was increased to 3188, which was close to the number of labels for the standing posture.



(**a**) (**b**) (**c**)

Figure 5. Schematic of simplified copy-paste: (**a**) crops from image with sitting posture; (**b**) background image of empty pen; (**c**) image with 4 sitting posture labels after copy-paste is performed.

2.6. Label Smoothing

Label smoothing, also known as label smoothing regularization, is a simple regularization technique that improves the generalization performance and accuracy of models in classification tasks, alleviating the class imbalance problem. The label smoothing is calculated as follows:

$$y(i) = \begin{cases} (1 - \text{factor}), \text{if}(i = \text{target}) \\ \frac{\text{factor}}{\text{Total categories}}, \text{if}(i \neq \text{target}) \end{cases} \quad (1)$$

In multi-classification tasks, real labels are often used in one-hot encoding. The neural network outputs a confidence level score corresponding to the current data for each class and normalizes these scores using SoftMax, which results in the probability that the current data belong to each class. Subsequently, the neural network promotes itself to learn more correct labels and eliminate incorrect labels by calculating the cross-entropy loss function. Over-fitting occurs when there are fewer training data covering all the sample characteristics to guarantee the generalization ability of the model. Label smoothing can solve the above problems, by soft one-hot encoding, to add noise, reduce the weight of real sample labels in calculating the loss function, and reduce the gap between the maximum and minimum values in the model prediction label. Consequently, there is the largest probability for positive samples and the smallest probability for negative samples, suppressing the effect of over-fitting.

2.7. Model Training Parameters

The experimental platform was a Windows 10 64-bit operating system. The hardware used in the experiment was an Intel(R) Core (TM) i5-10300H 16GB CPU and NVIDIA GTX 1660 Ti 6GB GPU. The programming language was Python 3.8 and the environment was CUDA-Toolkit 10.2. The PyTorch 1.7.1 (Facebook AI Research, Menlo Park, CA, USA) deep learning framework was built to conduct model building, training, and testing.

In this experiment, the model parameters were fine-tuned using random gradient descent, momentum, and learning rate warming-up strategies. The batch size of the model was set to 8 and a total of 150 epochs were trained; the initial learning rate was set to 0.01; the final learning rate was set to 0.002; the momentum was set to 0.937; the weight decay coefficient was set to 0.0005. The first three epochs were warmed up for the learning rate; the initial momentum for learning rate warming-up was 0.8, and the initial deviation learning rate for learning rate warming-up was set to 0.1.

2.8. Evaluation Metrics

To verify the validity of the methods presented in this paper, the following six widely used evaluation metrics for object detection were used in this experiment: precision, recall, average precision, mean average precision, speed, and model size.

Intersection over union (IoU) measures the degree of overlap between two bounding boxes. IoU calculates the ratio of the intersection and union of the prediction box and the ground truth box to measure the difference between the prediction box and the ground truth box. It is an additional parameter used to calculate the evaluation metrics. When the IoU is larger than the threshold set, the prediction box is considered correct. For a ground truth box and a prediction box, the IoU is calculated as follows:

$$\mathrm{IoU} = \frac{\mathrm{area}(\mathrm{box}_{\mathrm{pre}} \cap \mathrm{box}_{\mathrm{gt}})}{\mathrm{area}(\mathrm{box}_{\mathrm{pre}} \cup \mathrm{box}_{\mathrm{gt}})} \tag{2}$$

Precision is the ratio of the correct number of prediction boxes in a class to the total number of prediction boxes in that class. The formula is as follows, where TP is the number of prediction boxes with IoU larger than or equal to the set threshold and FP is the number of prediction boxes with IoU less than the set threshold.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3}$$

Recall refers to the ratio of the number of correct prediction boxes in a class to the total number of ground truth boxes in that class. The formula is as follows, where FN is the number of ground truth boxes not detected.

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4}$$

Average precision (AP) refers to the approximation of the area under a certain class of Precision-Recall curve, which is a value between 0 and 1. In practice, the Precision-Recall curve is smoothed; that is, for each point on the Precision-Recall curve, the precision value takes the maximum precision value on the right side of the point. The AP calculation formula is as follows:

$$\mathrm{AP} = \int_0^1 \mathrm{P}_{\mathrm{smooth}}(\mathrm{R})\mathrm{dR} \tag{5}$$

In this study, two AP values were used. One is $\mathrm{AP}_{0.5}$ with a fixed IoU threshold of 0.5. Another is $\mathrm{AP}_{0.5–0.95,}$ which means that the threshold value of IoU is adjusted from a fixed 0.5 to the value of AP every 0.05 in the interval 0.5–0.95, and the average value of all results is taken as the final result.

Mean average precision (mAP) refers to the mean of the average precision for different classes. The calculation formula is as follows. There are the following two corresponding mAPs: $\mathrm{mAP}_{0.5}$ and $\mathrm{mAP}_{0.5–0.95}$.

$$\mathrm{mAP} = (\sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{AP_i})/\mathrm{m} \tag{6}$$

Speed refers to the time cost to process a picture. The real-time performance of the model improves with a shorter average process time.

Model size refers to the size of model weight generated by model training, which is determined by the model architecture parameters.

3. Results and Discussion

3.1. Model Performance in Position Detection and Posture Recognition

This section mainly reports the model's performance after adding the new changes from YOLOv5s to YOLOXs. Table 3 shows the results of YOLOv5s, YOLOXs, and YOLOsx

in position detection. YOLOv5s and YOLOXs presented the same performance for pig position detection, but YOLOXs's speed was 1 ms slower and the model size increased by 1.9 M. The detection accuracy of YOLOsx was better than that of YOLOv5s and YOLOXs, with an increase of 0.2% in $mAP_{0.5–0.95}$, but the model size and speed were slightly inferior to those of YOLOv5s. This indicates that both YOLOv5s and YOLOXs show excellent position detection performance during the day and the night. YOLOsx may only achieve a small improvement in position detection because the evaluation metrics almost reach their highest values.

Table 3. Results of YOLOv5s, YOLOxs, and YOLOsx in position detection.

Model	Precision (%)	Recall (%)	$AP_{0.5}$ (%)	$AP_{0.5–0.95}$ (%)	Speed (ms)	Model Size(M)
YOLOv5s	99.8	99.5	99.5	90.8	8.5	14.4
YOLOXs	99.8	99.5	99.5	90.8	9.5	16.3
YOLOsx	99.8	99.6	99.5	91.0	9.5	16.2

Table 4 shows that the results of YOLOsx compared with the results of YOLOv5s and YOLOXs in posture recognition. Compared with YOLOv5s and YOLOXs, YOLOsx had the best detection accuracy. Its $mAP_{0.5}$ was 0.2% and 0.3% higher, and the $mAP_{0.5–0.95}$ was 0.4% and 0.3% higher than YOLOv5s and YOLOXs, respectively. YOLOsx was inferior to YOLOv5s in speed and model size, but the speed difference of 1 ms is satisfactory to meet the requirements in practical applications. Compared with YOLOX, the speed of YOLOsx did not slow down and the model size was reduced by 0.1 M, indicating that the new improved method of YOLOv5s can improve the detection performance of the model after being added to YOLOXs.

Table 4. Results of YOLOv5s, YOLOxs, and YOLOsxlc in posture recognition.

Model	Precision (%)	Recall (%)	$mAP_{0.5}$ (%)	$mAP_{0.5–0.95}$ (%)
YOLOv5s	95.0	90.0	94.3	85.4
YOLOX	93.2	92.5	94.2	85.6
YOLOsx	93.8	92.1	94.5	85.8

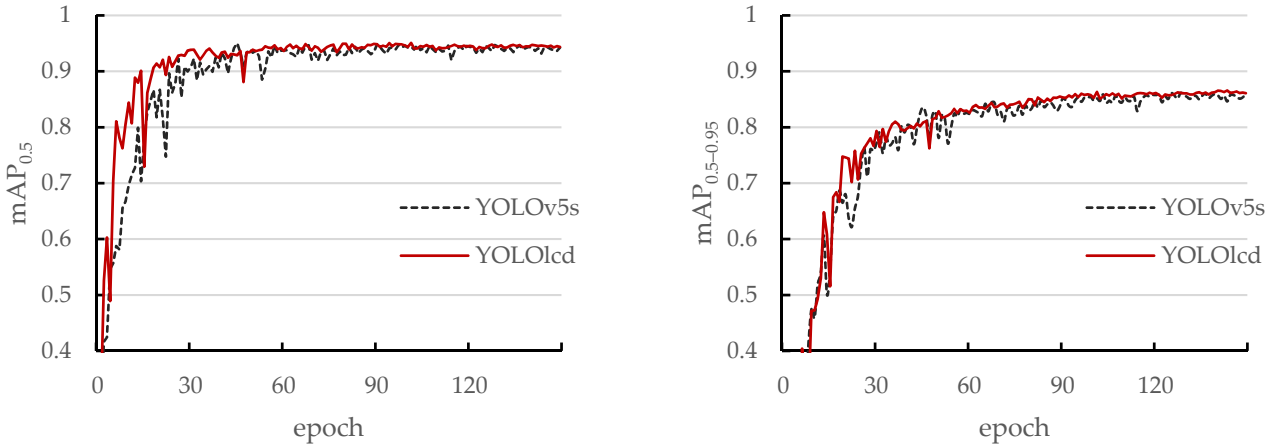
Although YOLOsx was reduced in speed and increased in model size compared with YOLOv5s, the gap was small, and it can meet the needs of the actual application. At the same time, the detection accuracy of YOLOsx was better than that of YOLOv5s and YOLOXs. Therefore, YOLOsx was used as the baseline for subsequent tests in this research.

3.2. The Results of Label Smoothing and Copy-Paste Added to YOLOsx for Posture Recognition

This section describes the experimental results of YOLOsx for pig posture recognition after adding label smoothing and copy-paste. For ease of description, YOLOsx + label smoothing is abbreviated to YOLOsxl; YOLOsx + copy-paste is abbreviated to YOLOsxc; YOLOsx + label smoothing + copy-paste is abbreviated to YOLOsxlc. After each training epoch, a validation step was carried out on the validation set, and the model weight with the best performance on the validation set was saved. The weight was also tested on the test set to compare the models' performance.

Figure 6 shows that the changes in YOLOsx and YOLOsxlc in the validation set during training. To present a comparison of the converged model, the size of the ordinate axis was adjusted so that $mAP_{0.5}$ and $mAP_{0.5–0.95}$ did not increase from the minimum value of the ordinate. It can be seen from the figure that in $mAP_{0.5}$, YOLOsxlc more easily causes the model to converge than YOLOsx. After the 25th epoch, YOLOsxlc remained above 0.9, while YOLOsx reached this value after 30 epochs. The convergence speed of YOLOsxlc on $mAP_{0.5–0.95}$ was also slightly faster than that of YOLOsx. Because the YOLOsx model is small, it shows a large oscillation before reaching convergence, which is slowed down on YOLOsxlc but still reflected. It may be that YOLOsxlc converges faster, thus reducing the

oscillation interval. After the models reached convergence, the performance of YOLOsxlc and YOLOsx on $mAP_{0.5}$ and $mAP_{0.5–0.95}$ was close, and only slightly improved.



(**a**) (**b**)

Figure 6. Changes in the mAP of YOLOsx and YOLOsxlc on validation set during training: (**a**) changes in $mAP_{0.5}$ on validation set; (**b**) changes in $mAP_{0.5–0.95}$ on validation set.

Table 5 shows the recognition results of standing, lying, sitting, and all postures after adding label smoothing and copy-paste to YOLOsx. It can be seen that YOLOsxl, YOLOsxc, and YOLOsxlc were better than YOLOsx in each type of posture recognition, which indicates that copy-paste and label smoothing can improve the detection accuracy of the model. In the standing and lying position recognition, YOLOsxl and YOLOsxc were better than YOLOsxlc at $AP_{0.5}$, while YOLOsxlc was better than YOLOsxl and YOLOsxc at $AP_{0.5–0.95}$. Higher $AP_{0.5–0.95}$ means that the model has better detection performance under the condition of a higher IoU threshold, so the combined use of copy-paste and label smoothing can improve the recognition accuracy of standing and lying positions. YOLOsxlc performed the best in detecting pigs' sitting posture, as well as in all classes of detection results. Compared with YOLOsx, YOLOsxl, and YOLOsxc, YOLOsxlc was 3.3%, 2%, and 1.4% higher on $AP_{0.5}$, and 2.3%, 1.8%, and 0.7% higher on $AP_{0.5–0.95}$. It showed that copy-paste and label smoothing had obvious effects on improving sitting posture recognition, and the combination of the two can achieve a greater improvement. In terms of speed and model size, there was no change because the model structure was not changed, and no parameters were added.

Table 5. Experimental results of copy-paste and label smoothing added to YOLOsx on the test set.

Classes	Metrics	YOLOsx	YOLOsxl	YOLOsxc	YOLOsxlc
Standing	Precision (%)	96.2	95.3	97.4	96.2
	Recall (%)	96.1	97.2	95.8	96.6
	$AP_{0.5}$ (%)	97.5	97.6	97.8	97.5
	$AP_{0.5–0.95}$ (%)	87.2	87.9	88.1	88.4
Lying	Precision (%)	97.5	96.8	97.8	97.5
	Recall (%)	98.5	99.0	98.1	98.6
	$AP_{0.5}$ (%)	98.5	98.7	98.7	98.7
	$AP_{0.5–0.95}$ (%)	89.7	90.2	90.3	90.5
Sitting	Precision (%)	87.7	85.9	88.7	88.5
	Recall (%)	81.8	85.2	83	86.7
	$AP_{0.5}$ (%)	87.6	88.9	89.5	90.9
	$AP_{0.5–0.95}$ (%)	80.5	81.0	82.1	82.8

Table 5. *Cont.*

Classes	Metrics	YOLOsx	YOLOsxl	YOLOsxc	YOLOsxlc
All	Precision (%)	93.8	92.7	94.6	94.1
	Recall (%)	92.1	93.8	92.3	93.9
	$mAP_{0.5}$ (%)	94.5	95.1	95.3	95.7
	$mAP_{0.5–0.95}$ (%)	85.8	86.4	86.9	87.2
	Speed (ms)	9.5	9.5	9.5	9.5
	Model size (M)	16.2	16.2	16.2	16.2

The above results show that copy-paste and label smoothing were effective in improving the model performance of YOLOsx. Increasing the number of sitting postures directly by copy-paste can provide YOLOsxlc with more opportunities to learn the characteristics of sitting postures. Label smoothing can reduce the sample weight of model error detection of sitting postures to other postures and alleviate the over-fitting of the model. Their improvement effects are similar. Therefore, the improvement of YOLOsxlc in sitting posture detection after the combination is also obvious.

3.3. Comparison of YOLOsxlc with Other Object Detection Models in Posture Recognition

To verify the effectiveness of the method proposed in this paper, the current typical two-stage and one-stage object detection models were compared with YOLOsxlc. Table 6 shows that the comparison results of YOLOsxlc and Faster R-CNN, SSD, FCOS, VarifocalNet, YOLOv3, YOLOXs, and YOLOv5s in pig posture recognition performance. YOLOsxlc had the highest detection accuracy on $mAP_{0.5}$ and $mAP_{0.5–0.95}$, especially on $mAP_{0.5–0.95}$, for which it was 18.2% higher than that of the lowest model, SSD. In terms of speed, YOLOsxlc was only 1 ms slower than YOLOv5s and was 108.7 ms faster than the slowest model, VarifocalNet. In terms of model size, YOLOsxlc was only 1.8 M larger than YOLOv5s, and it was 452.8 M smaller than the largest model, YOLOv3. In summary, the method proposed in this paper can effectively improve the detection accuracy for pig posture recognition while increasing the model size and speed less.

Table 6. Comparison of performance of different object detection models.

Model	$mAP_{0.5}$ (%)	$mAP_{0.5–0.95}$ (%)	Speed (ms)	Model Size (M)
Faster R-CNN	91.4	77.6	116.9	315
SSD	88.1	69.0	32.5	183
FCOS	91.2	79.0	107.1	244
VarifocalNet	90.9	78.1	118.2	249
YOLOv3	90.9	79.2	41.9	469
YOLOXs	94.2	85.6	9.5	16.3
YOLOv5s	94.3	85.4	8.5	14.4
YOLOsxlc	95.7	87.2	9.5	16.2

3.4. Visual Comparison of YOLOv5s, YOLOXs, and YOLOsxlc Detection Results

Figure 7 shows that the detection performance of YOLOv5s, YOLOXs, and YOLOsxlc during the day and night. The prediction box for the pig was generated by the model, and the class and confidence of the pig's posture were displayed on the prediction box. The confidence was used to judge whether the pig's posture in the prediction box was a positive sample or a negative sample. If the confidence value was less than the confidence threshold, this object was judged as a negative sample and background.

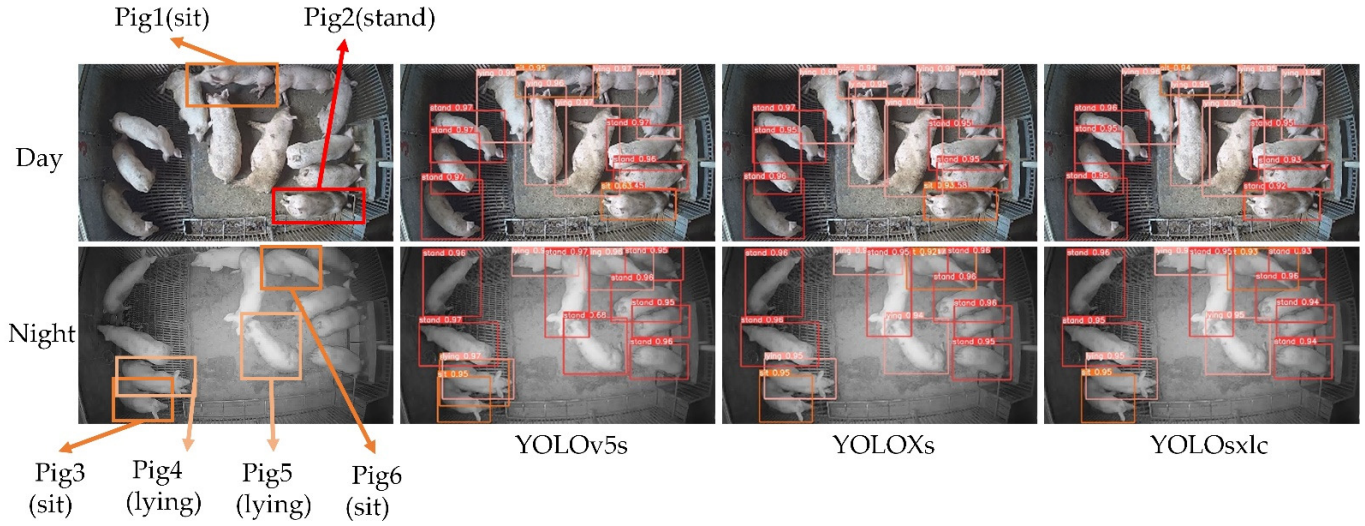


Figure 7. Comparison results of position detection and posture recognition between YOLOv5s, YOLOXs, and YOLOsxlc during daytime and nighttime.

The upper image in Figure 7 is an image of the pen taken during the day, with pig1 in a sitting posture and pig2 in a standing posture. YOLOXs mistakenly detected pig1 as lying, while YOLOv5s and YOLOsxlc correctly detected pig1 as sitting. In the lower right corner of the YOLOv5s and YOLOXs detection results, the model incorrectly detected the standing posture of pig2 as sitting, and YOLOsxlc correctly detected that pig2 was standing. It is concluded that the detection performance of YOLOsxlc during the day is better than that of YOLOv5s and YOLOXs.

The bottom image in Figure 7 was taken at night, with pig3 and pig6 in a sitting posture and pig4 and pig5 in a lying posture. Pig3 was in the lower left corner of the image, and all three models detected this posture correctly. Pig6 was in the upper right corner of the image, YOLOv5s misdetected pig3 as lying, and both YOLOXs and YOLOsxlc detected it correctly. YOLOv5s misdetected pig4 as sitting, and the prediction box of lying and sitting appeared on pig4 at the same time. The same phenomenon also occurred in the detection results of YOLOXs. The prediction box for the lying position also appeared on pig6. Under the same IoU threshold setting, this phenomenon was caused by the inaccurate recognition of the lying and standing postures by the model. In the YOLOv5s detection results, pig5 was wrongly detected as standing, while YOLOXs and YOLOsxlc both detected the lying posture correctly. The above results show that the detection performance of YOLOv5s was poor at night, and YOLOXs can improve the detection performance somewhat, but YOLOXs performed insufficiently in distinguishing lying postures and sitting postures without filtering redundant prediction boxes. YOLOsxlc can learn more features of pig sitting posture when enhanced by copy-paste, thus allowing the model to recognize more pig postures correctly.

The above results show that YOLOsxlc had better detection results during both day and night. They also prove that YOLOsxlc had a better detection effect on crowded, occluded, and low-visibility pigs, and thus it can meet the needs of pig posture recognition in real scenes.

3.5. Discussion

In this study, we improved the detection efficiency of deep learning in pig posture recognition, especially for pig sitting posture recognition, because the camera provided an overhead view of the dataset, and there was no further change in the number of pigs. Although the generalization performance of YOLOv5s and YOLOXs was improved after data augmentation, if more pigs under different camera views were detected, the learning ability of YOLOv5s and YOLOXs may be insufficient, resulting in a decline in the position

detection results. The learning ability of YOLOsxlc in posture recognition was better than that of YOLOv5s and YOLOXs, so YOLOsxlc may yield better results for position detection in more complex scenes.

In previous studies, Nasirahmadi et al. [27] used the same top view to recognize pigs standing, lying on their sides, and lying on their stomachs, and the maximum mAP reached 93%. The method proposed in this paper achieved a result of 2.5% higher if the sitting posture was included, and 5.3% higher if the sitting posture was not included, which proves the effectiveness of the method. Other researchers performed posture recognition from different camera views with 2D cameras [28–30], but the detection of sitting postures in pigs was not included in their experiments. It can be inferred that the recognition effect for sitting postures in the unmodified model shown in Table 5 was far lower than that for standing and lying postures. Therefore, if models from the previous study directly recognize the sitting posture of pigs, the detection effect may decline. In the research of Riekert et al. [28,29], Faster-RCNN was used to detect pig postures. As shown in Table 6, the speed of Faster-RCNN was lower than that of the method used in this paper. In the research of Shao et al. [30], YOLOv5 and DeeplabV3 were used for posture recognition. This additionally increased the cost of manual instance segmentation labels of pig postures.

In this paper, the simplified copy-paste method was applied as a data augmentation method for the dataset. It was proven that the detection results could be improved by only copying and pasting the bounding box of the object without additional instance segmentation annotation. This method can solve the problem of class imbalance caused by the lack of a class in the dataset. It can also be applied to other datasets with the same problem. The label smoothing strategy alleviated the over-fitting of the model by adding the labels to which the predicted value of the noise reduction model was too biased to the class with high probability. In the dataset of this paper, the model could not learn enough features because of the lack of sitting postures, which caused the prediction probability of sitting postures to be low. The label smoothing allowed the sitting posture to obtain a higher prediction probability, increased the generalization performance of the model and improved the overall effect of the model. As shown in Figure 4, the performances of YOLOsxlc and YOLOsx in the validation set were similar after convergence, but YOLOsxlc performed better in the test set. It is possible that YOLOsx experienced some over-fitting problems due to the small amount of data, and YOLOsxlc alleviated this problem effectively. Meanwhile, YOLOsxlc was able to achieve convergence in advance and shorten the training time.

The above discussion shows that the method proposed in this paper can improve the detection effect in pig posture recognition using 2D cameras, especially for pig sitting postures, and can meet the needs of pig farming and improve the automation level of precision pig farming.

The method in this paper also has some limitations. The camera view used a conventional top view. For pig farms with a large pen area, more camera views are used to expand the field of vision, and the number of pigs to be detected also increases. Therefore, we will collect more pig posture images from multiple camera views and finetune the model to improve the generalization performance. Due to hardware limitations, only YOLOv5s and YOLOXs were used as the baseline in this paper, and these models were small. If these models were replaced with larger models, the detection effect of the models could be improved, and further improvements could be obtained with the methods used in this paper. Although the recognition results for sitting postures have been greatly improved, the recognition of sitting postures is still lower than that of standing and lying postures due to the lack of height information regarding the backs of the pigs. The sitting posture information learned by the model may depend more on the shape of the pig than on the relative position information of the pig's feet and back. This is a drawback of using 2D cameras to capture images. Therefore, the use of images captured by depth cameras for posture recognition may improve the detection results of the model without considering the cost.

4. Conclusions

In this paper, a computer vision system based on deep learning technology was applied to the position detection and posture recognition of grouped pigs. Firstly, a dataset including standing, lying, and sitting postures of pigs captured by 2D cameras in the daytime and nighttime in a pig barn was established. Secondly, the newly improved method in YOLOv5 was added to YOLOX to improve the detection performance of the model. Finally, simplified copy-paste and label smoothing were used to solve the class imbalance caused by the lack of sitting postures among pigs in the dataset. In the test, the modified YOLOsxlc had an $AP_{0.5}$ of 99.5% and an $AP_{0.5–0.95}$ of 91% in pig position detection, an $AP_{0.5}$ of 90.9% and an $AP_{0.5–0.95}$ of 82.8% in sitting posture recognition, and an $mAP_{0.5}$ of 95.7% and an $mAP_{0.5–0.95}$ of 87.2% in all postures' recognition. The experimental results showed that the simplified copy-paste and label smoothing strategy was an effective method for solving the class imbalance caused by the lack of a class in the dataset, and the recognition results for the sitting posture of pigs were particularly improved after its use. At the same time, both methods can be extended to other datasets facing the class imbalance problem. Therefore, the method proposed in this paper can meet the needs of pig position detection and posture recognition in the actual production of pig farms. It also compensates for the lack of automatic recognition of sitting postures among grouped pigs in the existing research and could help to improve the automation level in precision pig framing. Our study lays a foundation for further automatic recognition and analysis of pig behavior.

Author Contributions: Data curation, H.J.; Investigation, H.J.; Methodology, H.J., F.L., and J.Y.; Project administration, G.T.; Software, H.J. and J.Y.; Supervision, G.T.; Writing—original draft, H.J.; Writing—review and editing, J.Y., Y.Z., Y.W., and F.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Chongqing Technology Innovation and Application Development Project (grant number: cstc2021jscx-dxwtBX0006).

Institutional Review Board Statement: The study proposal was approved by the Laboratory Animal Ethical Committee of China Agricultural University (AW41211202-5-1).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, Q.; Xiao, D. A review of video-based pig behavior recognition. *Appl. Anim. Behabiour. Sci.* **2020**, *233*, 105146. [CrossRef]
2. Ringgenberg, N.; Bergeron, R.; Devillers, N. Validation of accelerometers to automatically record sow postures and stepping behaviour. *Appl. Anim. Behav. Sci.* **2010**, *128*, 37–44. [CrossRef]
3. Oliviero, C.; Pastell, M.; Heinonen, M.; Heikkonen, J.; Valros, A.; Ahokas, J.; Vainio, O.; Peltoniemi, O.A.T. Using movement sensors to detect the onset of farrowing. *Biosyst. Eng.* **2008**, *100*, 281–285. [CrossRef]
4. Martinez-Aviles, M.; Fernandez-Carrion, E.; Lopez Garcia-Baones, J.M.; Sanchez-Vizcaino, J.M. Early Detection of Infection in Pigs through an Online Monitoring System. *Transbound. Emerg. Dis.* **2017**, *64*, 364–373. [CrossRef]
5. Maselyne, J.; Saeys, W.; De Ketelaere, B.; Mertens, K.; Vangeyte, J.; Hessel, E.F.; Millet, S.; Van Nuffel, A. Validation of a High Frequency Radio Frequency Identification (HF RFID) system for registering feeding patterns of growing-finishing pigs. *Comput. Electron. Agric.* **2014**, *102*, 10–18. [CrossRef]
6. Borges Oliveira, D.A.; Ribeiro Pereira, L.G.; Bresolin, T.; Pontes Ferreira, R.E.; Reboucas Dorea, J.R. A review of deep learning algorithms for computer vision systems in livestock. *Livest. Sci.* **2021**, *253*, 104700. [CrossRef]
7. Li, G.; Huang, Y.; Chen, Z.; Chesser, G.D., Jr.; Purswell, J.L.; Linhoss, J.; Zhao, Y. Practices and Applications of Convolutional Neural Network-Based Computer Vision Systems in Animal Farming: A Review. *Sensors* **2021**, *21*, 1492. [CrossRef]
8. Lao, F.; Brown-Brandl, T.; Stinn, J.P.; Liu, K.; Teng, G.; Xin, H. Automatic recognition of lactating sow behaviors through depth image processing. *Comput. Electron. Agric.* **2016**, *125*, 56–62. [CrossRef]
9. Chen, C.; Zhu, W.; Norton, T. Behaviour recognition of pigs and cattle: Journey from computer vision to deep learning. *Comput. Electron. Agric.* **2021**, *187*, 106255. [CrossRef]

10. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*, 1st ed.; MIT Press: Cambridge, MA, USA, 2016; pp. 1–775. ISBN 978-0-262-03561-3.
11. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
12. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 580–587. [CrossRef]
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *Computer Vision–ECCV 2014, Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Springer: Cham, Switzerland, 2014; Volume 8691, pp. 346–361. [CrossRef]
15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788. [CrossRef]
16. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Springer: Cham, Switzerland, 2016; Volume 9905, pp. 21–37. [CrossRef]
17. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points. *arXiv* **2019**, arXiv:1904.07850.
18. Li, Y.; Sun, L.; Zou, Y.; Li, Y. Individual pig object detection algorithm based on Gaussian mixture model. *Int. J. Agric. Biol. Eng.* **2017**, *10*, 186–193. [CrossRef]
19. Sa, J.; Choi, Y.; Lee, H.; Chung, Y.; Park, D.; Cho, J. Fast Pig Detection with a Top-View Camera under Various Illumination Conditions. *Symmetry* **2019**, *11*, 266. [CrossRef]
20. Seo, J.; Ahn, H.; Kim, D.; Lee, S.; Chung, Y.; Park, D. EmbeddedPigDet-Fast and Accurate Pig Detection for Embedded Board Implementations. *Appl. Sci.* **2020**, *10*, 2878. [CrossRef]
21. Ahn, H.; Son, S.; Kim, H.; Lee, S.; Chung, Y.; Park, D. Ensemble PigDet: Ensemble Deep Learning for Accurate Pig Detection. *Appl. Sci.* **2021**, *11*, 5577. [CrossRef]
22. Yan, H.; Liu, Z.; Cui, Q.; Hu, Z. Multi-target detection based on feature pyramid attention and deep convolution network for pigs. *Trans. Chin. Soc. Agric. Eng.* **2020**, *36*, 193–202.
23. Fang, J.; Hu, Y.; Dai, B.; Wu, Z. Detection of group-housed pigs based on improved CenterNet model. *Trans. Chin. Soc. Agric. Eng.* **2021**, *37*, 136–144.
24. Zheng, C.; Zhu, X.; Yang, X.; Wang, L.; Tu, S.; Xue, Y. Automatic recognition of lactating sow postures from depth images by deep learning detector. *Comput. Electron. Agric.* **2018**, *147*, 51–63. [CrossRef]
25. Zhu, X.; Chen, C.; Zheng, B.; Yang, X.; Gan, H.; Zheng, C.; Yang, A.; Mao, L.; Xue, Y. Automatic recognition of lactating sow postures by refined two-stream RGB-D faster R-CNN. *Biosyst. Eng.* **2020**, *189*, 116–132. [CrossRef]
26. Yang, X.; Zheng, C.; Zou, C.; Gan, H.; Li, S.; Huang, S.; Xue, Y. A CNN-based posture change detection for lactating sow in untrimmed depth videos. *Comput. Electron. Agric.* **2021**, *185*, 106139. [CrossRef]
27. Nasirahmadi, A.; Sturm, B.; Edwards, S.; Jeppsson, K.; Olsson, A.; Mueller, S.; Hensel, O. Deep Learning and Machine Vision Approaches for Posture Detection of Individual Pigs. *Sensors* **2019**, *19*, 3738. [CrossRef]
28. Riekert, M.; Klein, A.; Adrion, F.; Hoffmann, C.; Gallmann, E. Automatically detecting pig position and posture by 2D camera imaging and deep learning. *Comput. Electron. Agric.* **2020**, *174*, 105391. [CrossRef]
29. Riekert, M.; Opderbeck, S.; Wild, A.; Gallmann, E. Model selection for 24/7 pig position and posture detection by 2D camera imaging and deep learning. *Comput. Electron. Agric.* **2021**, *187*, 106213. [CrossRef]
30. Shao, H.; Pu, J.; Mu, J. Pig-Posture Recognition Based on Computer Vision: Dataset and Exploration. *Animals* **2021**, *11*, 1295. [CrossRef]
31. Beattie, V.E.; Walker, N.; Sneddon, I.A. Effects of environmental enrichment on behavior and productivity of growing pigs. *Anim. Welf.* **1995**, *4*, 207–220.
32. Jarvis, S.; Calvert, S.K.; Stevenson, J.; Vanleeuwen, N.; Lawrence, A.B. Pituitary-adrenal activation in pre-parturient pigs (Sus scrofa) is associated with behavioural restriction due to lack of space rather than nesting substrate. *Anim. Welf.* **2002**, *11*, 371–384.
33. Guo, Y. *The Bahavioural Characteristics Research of Fattening Pigs Reared in Thedeep-Litter System and the Concretefloor System with Outdoor Runs*; Nanjing Agricultural University: Nanjing, China, 2017.
34. Guo, Y.; Lian, X.; Yan, P. Diurnal rhythms, locations and behavioural sequences associated with eliminative behaviours in fattening pigs. *Appl. Anim. Behav. Sci.* **2015**, *168*, 18–23. [CrossRef]
35. Brunberg, E.; Wallenbeck, A.; Keeling, L.J. Tail biting in fattening pigs: Associations between frequency of tail biting and other abnormal behaviours. *Appl. Anim. Behav. Sci.* **2011**, *133*, 18–25. [CrossRef]
36. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2019; pp. 10778–10787. [CrossRef]

37. Technology, Z.G.S.L. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
38. Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.; Cubuk, E.D.; Le, Q.V.; Zoph, B. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation. *arXiv* **2020**, arXiv:2012.07177.