Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 7078670, 16 pages
https://doi.org/10.1155/2022/7078670

Research Article

Detection and Counting Method of Pigs Based on YOLOV5_Plus: A Combination of YOLOV5 and Attention Mechanism

Zishun Zhou

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610000, China

Correspondence should be addressed to Zishun Zhou; 2019110102030@std.uestc.edu.cn

Received 15 April 2022; Revised 16 June 2022; Accepted 29 June 2022; Published 22 August 2022

Academic Editor: Saeid Jafarzadeh Ghoushchi

Information-based pig detection and counting is the trend in smart animal husbandry development. Cameras can efficiently collect farm information and combine it with artificial intelligence technology to assist breeders in real-time monitoring and analysis of farming. In order to improve the speed and accuracy of pig detection and counting, an advanced improved YOLO_v5 method for pig detection and counting based on the attention mechanism is proposed. The model is named as YOLOV5_Plus. This article utilizes a series of data augmentation methods, including translation, color augmentation, rescaling, and mosaic. The proposed model performs feature extraction on the original image with a backbone network, detects pigs of different sizes with three detection heads, and counts the detected anchor frames. Different versions of YOLOV5 are compared, and YOLOV5x is selected as the baseline model for the best performance. Attention modules are smartly combined with the model so that the model can better handle overlapping and misidentification. YOLOV5_Plus can achieve an accuracy of 0.989, a recall of 0.996, mAP@.50 of 0.994, and mAP@.50:.95 of 0.796, which outperforms all competing models. The inference time per image during detection is only 24.1 ms. YOLOV5_Plus model achieves real-time pig number and location detection, which is meaningful for promoting smart animal husbandry and saving labor costs in farming enterprises.

1. Introduction

Pork is one of the most important food sources for humans. According to the food and Agriculture Organization of the United Nations (FAO) [1], pork is rich in proteins, amino acids, and vitamins B6 and B12, which are the main ingredients of many traditional diets. In recent years, farm breeding has been expanding, and more and more farms want to monitor farm conditions through digital means accurately. For instance, real-time and accurate monitoring of the number of pigs can help farmers monitor the feeding density scientifically and rationally. In addition, pig counting can assist in accurate feeding, avoid feed waste, improve feed conversion rate, and increase economic efficiency.

Pig counting is facing many difficulties in large-scale agricultural production management. The traditional pig counting method relies on the visual observation of the breeder. Due to the large size of pig herds and overlap, manual counting of the number of pigs is easy to miss, which is both time-consuming and costly. Pig counting based on deep learning also faces several challenges. For instance, light variations can have an impact on pig detection. Different conditions such as side light, strong light, and backlight can cause more intraclass variance. The varying postures of pigs also make object detection more challenging. Besides, a large scale of overlapping can easily result in low recall in the detection process.

With the rapid development of computer hardware and related theories, deep learning and deep neural networks have significantly improved in accuracy and speed in the past few years. Among them, the progress of target detection algorithms in deep learning is particularly impressive. It mainly includes one-stage algorithms and two-stage algorithms. One-stage algorithms include YOLOV1 [2], SSD [3], YOLOV2 [4], YOLOV3 [5], YOLOV4 [6], YOLOV5, etc. Two-stage algorithms include R-CNN [7], Fast R-CNN [8], Faster R-CNN [9], Mask R-CNN [10], etc. Deep learning-

based target detection algorithms have been widely used in agriculture and animal husbandry. Many algorithms have been designed to be applied to target counting, tracking, and other upper-layer applications. For example, a two-route convolutional neural network (CNN) was proposed by Ramin et al. [11] to detect and classify COVID-19 infection from CT images. To improve classification accuracy, fuzzy c-means clustering and local directional pattern (LDN) encoding methods were used to represent the input image, respectively, to find more complex patterns in the image. In the area of separation of breast cancer lesions, Jafarzadeh Ghoushchi et al. [12] designed a well-designed CNN consisting of an autoencoded stacking (SAE) model with a logistic regression layer at the top of the network to monitor the flow. They applied CNNs, VGGs, and residual networks, respectively, to the Breast Cancer Database (BCDR-DM), and the result showed that CNNs outperformed the other two models. Yu et al. [13] proposed a deep learning network model based on multi-modules and attention mechanism (MAN) to realize the counting of cultured fish, which consists of a feature extraction module, attention module, and density map estimation module. Among them, the feature extraction module was composed of three parallel convolutional networks, which are used to extract the general feature map of the image and serve as the input of the subsequent module. The density map estimation module represents the distribution and the number of fishes in the image. The experimental results for MAN showed that the counting accuracy is about 97.12% and the deviation is 3.67. In the point pattern analysis based on YOLO object detection algorithms, Petso et al. [14] treated each animal as a point and identified five animal species by the behavioral pattern of those points. Animal features are harder to detect at higher altitudes and in the presence of environmental camouflage, animal occlusion, and shadows. The point pattern algorithms produced an $F1$-score above 96% across all drone altitudes.

Pig detection and counting are essential for accurate and fast counting of livestock. Ahrendt et al. [15] designed a computer vision system based on support maps to track loose housed pigs. It can achieve tracking of at least three pigs at the same time. Tian et al. [16] proposed a new solution for pig counting on the farm using deep learning. The network they designed was based on the combination of counting CNN and ResNeXt model, which achieved an improved high accuracy with low computational cost. The results demonstrated that in real-world data, this method got a mean absolute error of 1.67. Riekert et al. [17] used a deep learning system to detect the position and pose of pigs and achieved 84% mAP@.50 for the day and 58% mAP@.50 for the night. Although scholars have studied pig identification, the accuracy and speed of detection for pig identification are not satisfactory. In addition, pig detection is the basis of pig behavior analysis. Based on the detection and tracking of the pig's location, it is possible to analyze its behavior. For example, Kashiha et al. [18] proposed an ellipse fitting algorithm based on image pattern recognition to detect the position of a pig in a pen and analyze the specific behavior of the pig. Pigs could be identified with an average accuracy of 88.7%. Although the abovementioned researchers have reported studies on pig identification, the accuracy rate for pig identification is still below 95%. Therefore, improving the accuracy of pig identification is still relevant, especially in highly overlapping pig breeding environments.

However, no researcher has yet used YOLO-based algorithms for pig counting. YOLOV5, the latest generation of the YOLO family, is an advanced, fast, and accurate detector. Unlike density map-based counting methods, YOLOV5-based counting can directly detect the location and size of each target and then count the number of pigs based on the identified targets. It allows counting the number of pigs and annotating the pigs directly in the original image, which can better visualize the behavior of the pigs and facilitate movement detection. In recent studies, YOLOV5 has been used in various target detection tasks. Wang and Yan [19] used YOLOV5 for leaf detection and verified that the detection speed of YOLOV5 was significantly higher than Faster R-CNN with no significant difference in mAPs. Wang et al. [20] combined YOLOV5 and Siamrpn++ to propose a high-precision fish detection and tracking algorithm that achieves 76.7% accuracy and enables real-time tracking. Dong et al. [21] proposed a novel lightweight YOLOV5 network designed for vehicle detection. A convolutional block attention module (CBAM) was introduced to the backbone network to improve the model performance to select critical information, and CIoU_Loss was applied to the bounding box regression loss function to accelerate the bounding box regression rate. The proposed model achieved considerably better results compared to relevant methods. Given the excellent performance of YOLOV5 in terms of detection accuracy and speed, we decided to transfer YOLOV5 to the field of pig detection.

The rest of this article is organized as follows: Section 2 describes the data and algorithms used in the article, including data enhancement, the YOLO series of target detection algorithms, and the attention mechanism. Section 3 shows the model comparison results and validates the performance of the improved algorithm-YOLOV5_Plus. Section 4 discusses the advantages and limitations of the model. Section 5 summarizes the whole article and provides an outlook on prospects. The main contributions of this study are as follows:

(1) Color augmentation, translation, shear, rescaling, and mosaic processing for images taken in a pig feeding environment.

(2) To improve an improved YOLOV5 network architecture based on a convolutional block attention module (CBAM) for the problem of pig aggregation, which tends to lead to poor pig recognition accuracy.

2. Materials and Methods

2.1. Data Description. The dataset used in this experiment is public [22]. The original data files include 700 images in JPG format, and each photo contains multiple images of one pigpen. The annotation files include 700 files in JSON format. The annotation files and the photos are one-to-one

correspondence. The annotation information includes the coordinates of each pig's top left and bottom right corners of the box rectangle in the picture and the corresponding picture name. Figure 1 shows the samples of annotated pictures. The experiment randomly divided the dataset into training and validation sets of 549 and 151 images, respectively. The test set contains 220 images without annotation information. Compared with the training set, the test set includes more objects than pigs, mainly feeders, troughs, and light cords.

We convert the annotation information of the training set from json format to txt format. The converted annotation format is <id>, <center x>, <center y>, <width>, and <height>.

There are roughly 20 pigs in each image. All annotations were carefully checked manually to mark all pigs in the images.

2.2. Image Augmentation. Data augmentation can artificially expand the dataset, increase the diversity of the data, and improve the robustness of the model [23]. In recent years, it has been widely used in various research fields. Common data augmentation methods include geometric transformations such as cropping [24, 25], flipping [24–26], rotating [25, 26], scaling [24, 25], and warping [24, 25], as well as pixel scrambling [27], adding noise [25], illumination adjustment [25, 26], and contrast adjustment [25, 26]. Mosaic was first applied to YOLOV4 and can significantly improve the average precision (mAP). It stitches four images into one mosaic image, which can show more detection of target objects, speed up the effect of training the model, and improve the generalization of the model. Figure 2 shows the results of some augmented images. During the training process of this study, all data enhancement methods are turned on probabilistically.

2.3. Detection Principle

2.3.1. The Development of YOLO. YOLO was firstly proposed in 2016. Before that, two-stage algorithms usually consisted of two parts: (1) generating candidate regions and (2) classifying the candidate regions using a classification network. Although the two-stage algorithm can achieve high accuracy, the model is often complicated and large with low speed, which makes it hard to enable deployment on mobile devices. The YOLO algorithm transforms the target detection problem into a regression problem. The images are first divided into $S \times S$ grids, predicting B bounding boxes. Each grid point predicts the target whose centroid lies at that grid point, its confidence score, and C conditional category probabilities $pr(\text{class}_i|\text{object})$. The confidence equation is as follows:

$$\frac{\text{Confidence} = \Pr(\text{class}_i|\text{object}) \times \Pr(\text{object}) \times IOU_{\text{pred}}^{\text{truth}}}{\Pr(\text{object}) \in \{0, 1\}}, \tag{1}$$

where $\Pr(\text{object})$ is equal to 1 only if the grid contains one object.

The loss function calculates the sum of the position, width, height, and confidence errors of the prediction box with respect to the ground truth using the mean square error. The non-maximum suppression (NMS) is used to select the best bounding box if multiple bounding boxes detect the same target when detection is performed. YOLOV2 introduces the anchor mechanism, which uses k-means clustering to generate the width and height of the anchor to better match objects of different sizes. The backbone network of YOLOV2 is DarkNet-19, which is quite fast with fewer parameters. YOLOV3 makes some incremental improvements based on YOLOV2. It uses DarkNet-53 as the backbone network, extracting the most advanced techniques for target detection at that time, such as ResNet, DenseNet, and FPN. Compared with ResNet-152, DarkNet-53 has a faster speed (78 FPS compared to 37 FPS in ImageNet) and similar accuracy. YOLOV3 sets nine prediction boxes at each grid point and uses logistic regression to calculate the object score for each bounding box, ignoring the IOU samples more enormous than the threshold but not the best, which can significantly reduce the computational effort.

In YOLOV4, a new backbone network, CSPDarkNet53, is adopted, and the Mish activation function is used instead of LeakyReLU. In the neck network, PANet (path aggregation network) is used for feature fusion instead of FPN. Moreover, the spatial attention module (SAM), an attention mechanism, is introduced. In terms of data augmentation. YOLOV4 proposes the mosaic for the first time, where several images are cropped and stitched together to form new training set elements. These methods made YOLOV4 the state-of-the-art target detection algorithm at that time.

2.3.2. YOLOV5. The structure of YOLOV5 is similar to that of YOLOV4. It contains four main parts: input, backbone, neck, and head. The YOLOV5 used in this article is YOLOV5 v6.0, released on 12 October 2021. The structure of YOLOV5s is shown in Figure 3.

The input end includes mosaic data enhancement, image size processing, and adaptive anchor box calculation. Mosaic data enhancement enriches the background and the number of small objects in the dataset by combining four images. Image size processing adaptively adds minimum black borders to the original images of different lengths and widths and uniformly scales them to a standard size. The adaptive anchor box calculation compares the output predicted boxes with the real boxes based on the initial anchor boxes, calculates the gap, and then updates it in reverse, continuously iterating the parameters to obtain the most suitable anchor box value.

The backbone network of YOLOV5 mainly consists of BottleneckCSP(C3), Focus, and SPP (SPPF) modules. The original CSPNet [28] (Cross Stage Partial Network) splits the feature map into two parts. One part is convolved directly, and the other part goes through the dense block. Then the two parts are concatenated together. With CSPNet,

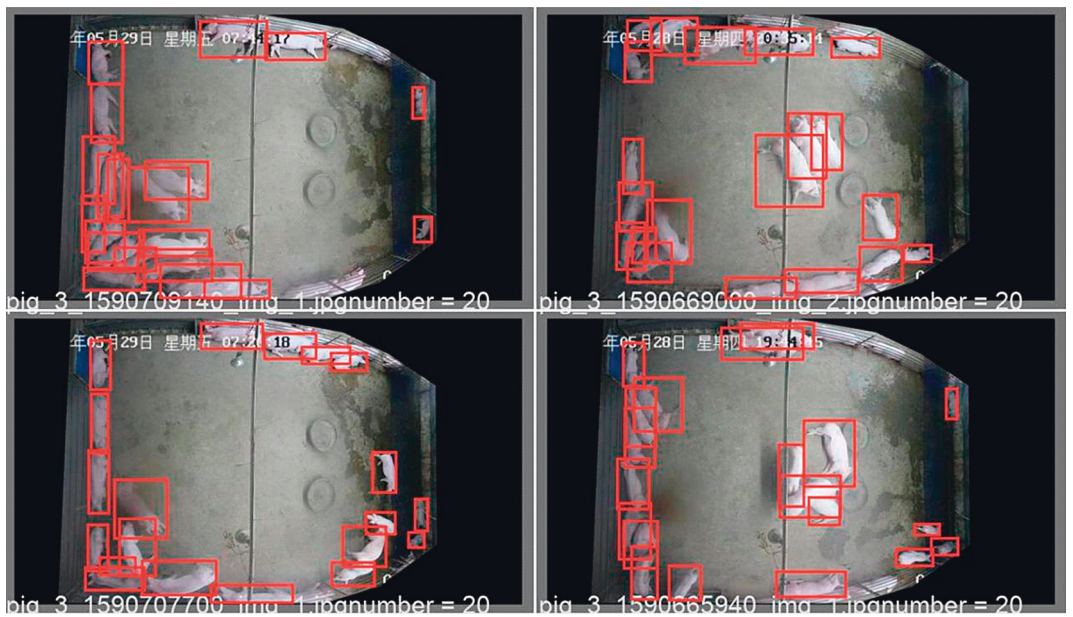
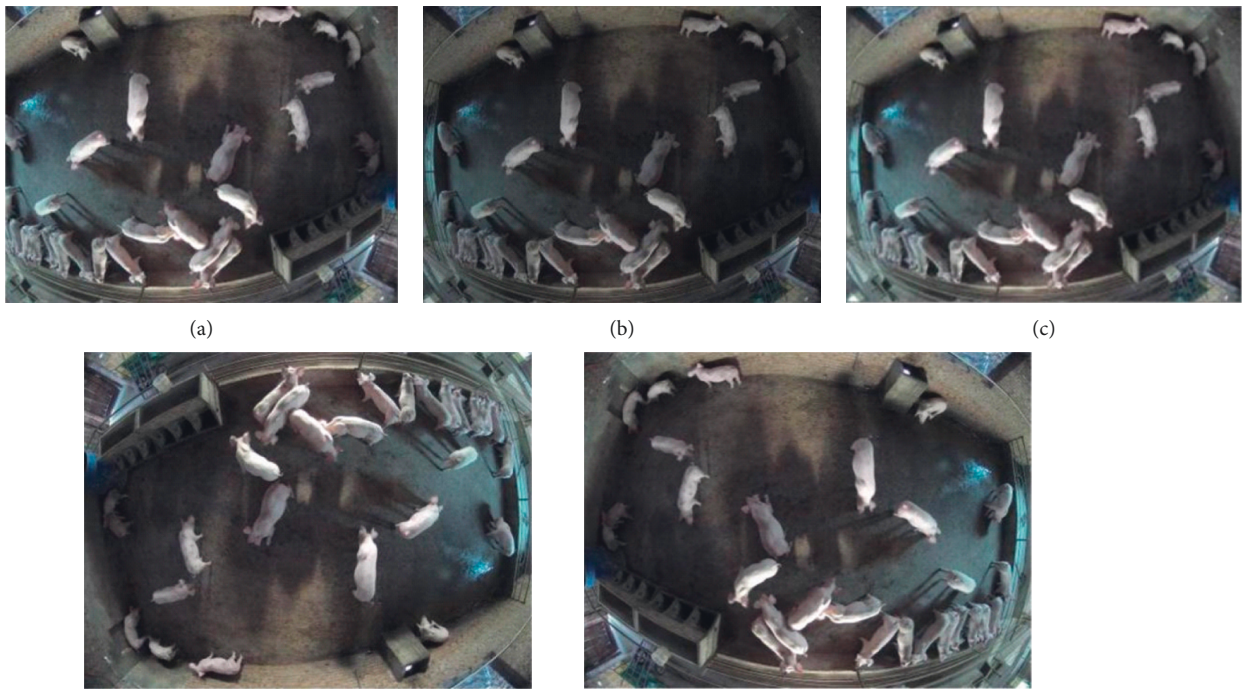


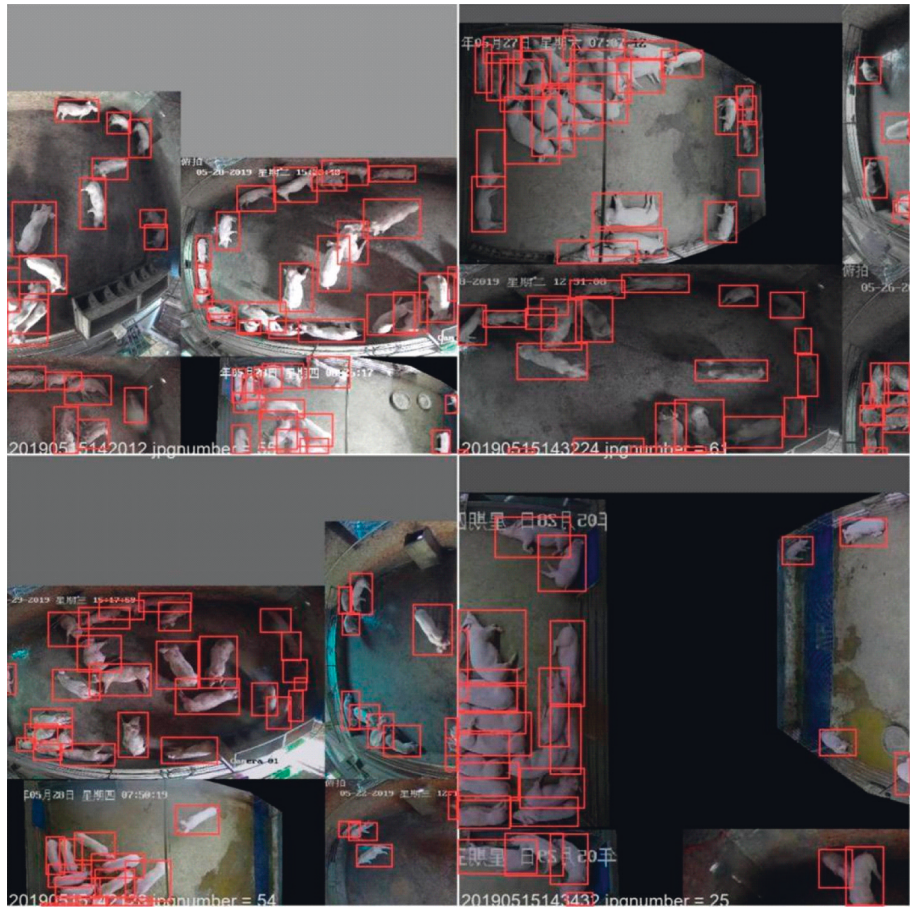
Figure 1: Annotation of pigs in the images.



(a) (b) (c)

(d) (e)

Figure 2: Continued.



(f)

Figure 2: Examples of image augmentation. (a) Translate. (b) Color augmentation. (c) Rescaling. (d) Flip up-down. (e) Flip right-left. (f) Mosaic augmentation.

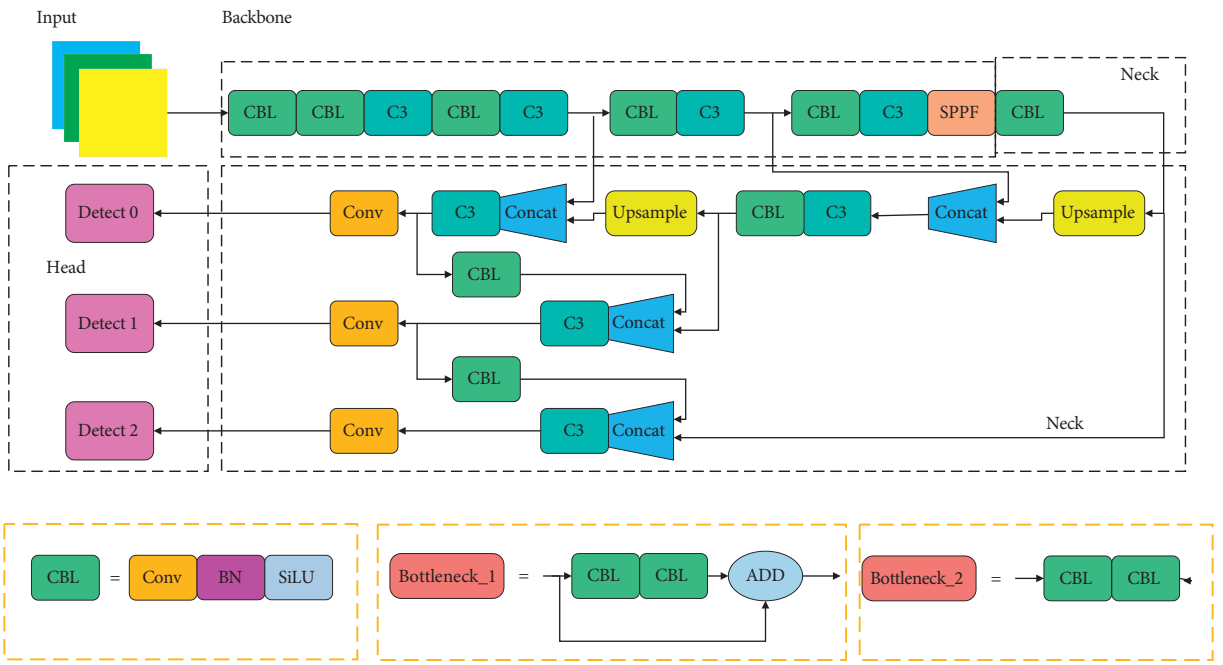


Figure 3: The structure of YOLOV5s.

computational costs can be greatly reduced, and accuracy can be improved to a certain extent. In the latest YOLOV5, the C3 module is mainly applied instead of BottleneckCSP. Compared with BottleneckCSP, C3 contains only three convolutions except for the Bottleneck part. It is simpler, faster, and lightweight with similar performance to BottleneckCSP. The bottleneck is one of the components of BottleneckCSP(C3). YOLOV5 uses two different bottlenecks in the backbone and neck: Bottleneck_1 and Bottleneck_2. The structure of both bottlenecks is shown in Figure 3. The structure of BotleneckCSP and C3 is shown in Figures 4 and 5, respectively.

The Focus module divides the input image data into four pieces. The four pieces are generated by changing the width and height to 1/2 of the original input data and the number of channels to four times, obtaining the two-fold downsampling feature map without information loss. The Focus module is usually located at the beginning of the backbone network. These four pieces of data are spliced together in the channel dimension and then convolved to obtain a binary downsampled feature map without information loss. The Focus module realizes downsampling while increasing the channel dimension, reducing FLOPs, and increasing speed. The operation of the focus slice is shown in Figure 6.

The SPP module is located at the end of the backbone. The spatial pyramid pooling module executes the maximum pooling with different kernel sizes and fuses the features by concatenating them. It combines different resolutions into the features by pooling the images with different sizes (kernel size = 5, 9, 13). A more lightweight structure SPPF is used in YOLOV5 v6.0 with all kernels of size 5. The structure of SPP and SPPF is shown in Figures 7 and 8, respectively.

YOLOV5 adopts the FPN and PAN structure in the neck. FPN is top-down, passing down the strong semantic features from the top layer to augment the pyramid. However, FPN only enhances the semantic information, not the localization information. PAN adds a bottom-up enhancement behind FPN. The feature map at the top layer can also enjoy the rich location information brought by the bottom layer, which improves the accuracy of the model. The structure of the neck is shown in Figure 9.

The head end outputs a vector with the category probability of the target object, the object score, and the position of the bounding box for that object. The detection network consists of three detection layers with different size feature maps used to detect target objects of different sizes. Each detection layer outputs the corresponding vector and finally generates the prediction bounding box and category of the object in the original image and marks it.

YOLOV5 contains four network structures due to different widths and depths: YOLOV5s, YOLO V5m, YOLOV5l, and YOLOV5x. YOLOV5s have the smallest width and depth with the fastest speed; YOLOV5x has the largest width and depth but runs relatively slower.

2.3.3. Convolutional Block Attention Module (CBAM). CBAM [29] was proposed by Woo et al. in 2018, which consists of two submodules: the channel attention module and the spatial attention module.

(1) Channel Attention Module. Channel attention focuses on exploring the relationship between feature maps of different channels. To enhance the representational power of the network, the channel attention module uses both average-pooled and max-pooled features, which are fed into a shared MLP with only one hidden layer to generate the channel attention map $M_C \in R^{C\times 1\times 1}$. The final output is obtained by fusing the two feature vectors through element-wise summation. The channel attention formulation is shown in equation (2). Figure 10 shows the structure of the spatial attention module

$$\begin{aligned} M_c(F) &= \sigma\left(MLP\left(\text{AvgPool}\left(F\right)\right) + MLP\left(\text{MaxPool}\left(F\right)\right)\right) \\ &= \sigma\left(\left(W_1 W_0 F^c_{\text{avg}}\right) + \left(W_1 W_0 F^c_{\text{max}}\right)\right). \end{aligned} \tag{2}$$

(2) Spatial Attention Module. Different from channel attention, spatial attention is mainly concerned with where the information is concentrated. The algorithm first performs the average-pooling and max-pooling operations on the channel-refined feature in the extended channel direction to obtain two feature maps with dimensions $H \times W \times 1$. Then these two feature maps are stitched together and performed a convolution operation to generate a spatial attention map $M_S(F) \in R^{H\times W}$. The formula of spatial attention is presented in Eq. (3). Figure 11 shows the structure of the spatial attention module.

$$\begin{aligned} M_S(F) &= \sigma\left(f^{7\times 7}\left(\left[\text{AvgPool}\left(F\right); \text{MaxPool}\left(F\right)\right]\right)\right) \\ &= \sigma\left(f^{7\times 7}\left(\left[F^S_{\text{avg}}; F^S_{\text{max}}\right]\right)\right). \end{aligned} \tag{3}$$

(3) Arrangement of Attention Modules. Previous research shows that sequential arrangement of channel attention and spatial attention modules can perform better than parallel arrangement. Figure 12 shows the overall structure of CBAM.

2.3.4. The Improved YOLOV5_Plus. YOLOV5_Plus is an improved model based on YOLOV5x. Previously, attention mechanisms have been tried in other fields combined with YOLOV5. For example, Yan et al. [30] combined the SE module with YOLOV5 to improve the accuracy of coal-gangue classification. Qi et al. [31] achieved high-accuracy recognition of tomato virus disease and improved detection speed based on a YOLOV5 and SE module model. However, how the attention mechanism can be applied to overlapping and dense target recognition and how it performs in the counting domain are unknown. Based on this, we combine the attention mechanism with YOLOV5 and conduct a series of experiments to find the best solution. YOLOV5_Plus can better extract channel and spatial features in the feature map by utilizing the attention mechanism.

The attention mechanism has been used in various target detection algorithms to obtain better results. Attention mechanisms can make models more robust to objects of different locations and sizes and avoid overfitting problems. For example, Ranjbarzadeh et al. [32] combined cascaded convolutional neural network and DWA (distance-wise

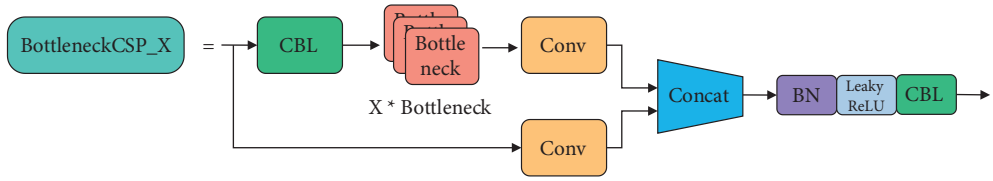


Figure 4: The structure of BottleneckCSP_X.

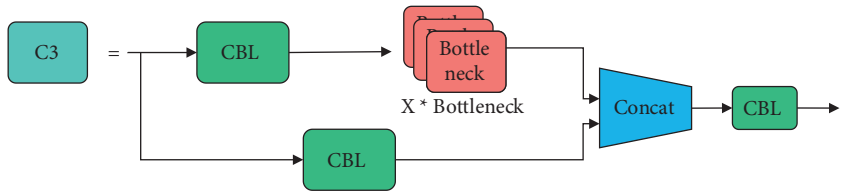


Figure 5: The structure of CSP2_X.

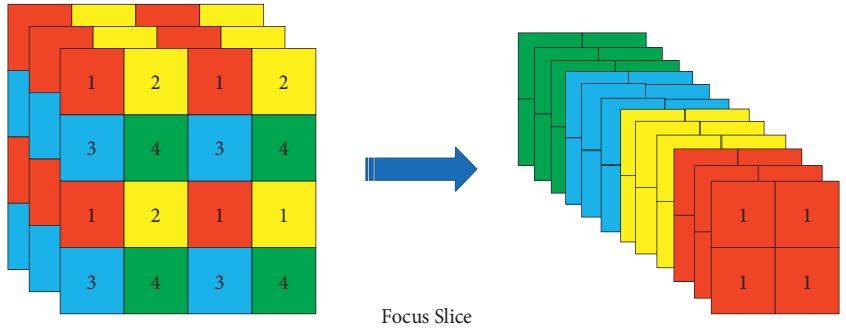


Figure 6: Focus slice operation.

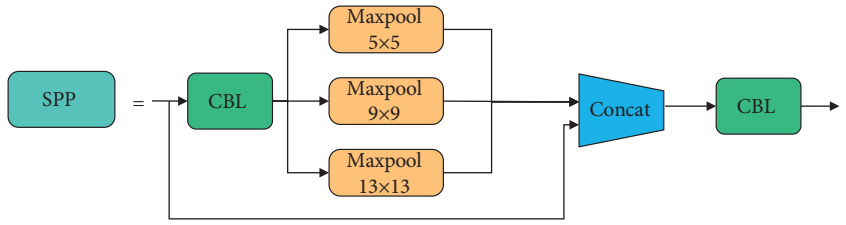


Figure 7: The structure of SPP.

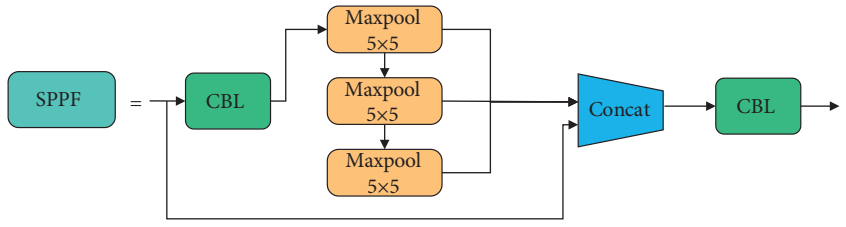


Figure 8: The structure of SPPF.

attention) to develop a new brain tumor segmentation architecture. The attention mechanism greatly improved the performance of the model by applying key location features of the image to the fully connected layer. This study carried out a comprehensive and systematic investigation of how the attention module was added. We experimented with different attention modules, including CBAM, SE [33], and CoordAtt [34]. Different ways of adding attention modules to the model have been investigated. One option is to replace the CBL module in the backbone with an attention module. It is also the way chosen by YOLOV5_Plus. Another way is to replace the CBL module in BottleneckCSP (C3) with the attention module. Figure 13 shows the improved network structure. From Figure 13(a), it can be seen that the four

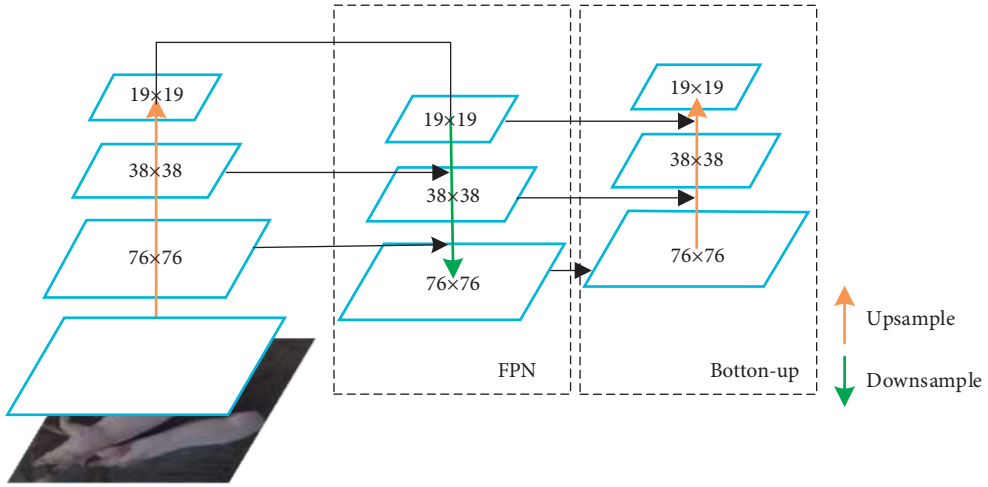


Figure 9: The structure of FPN and PAN.

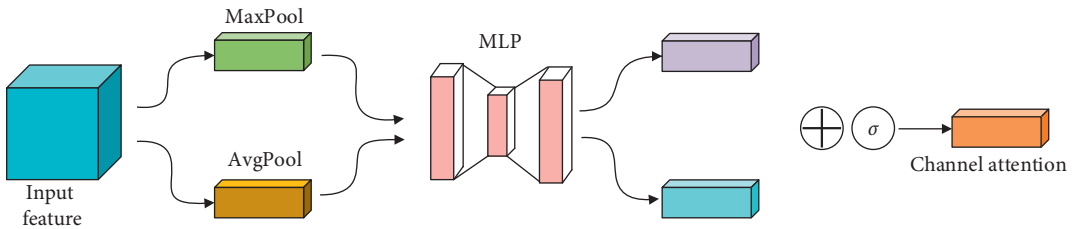


Figure 10: Channel attention module.

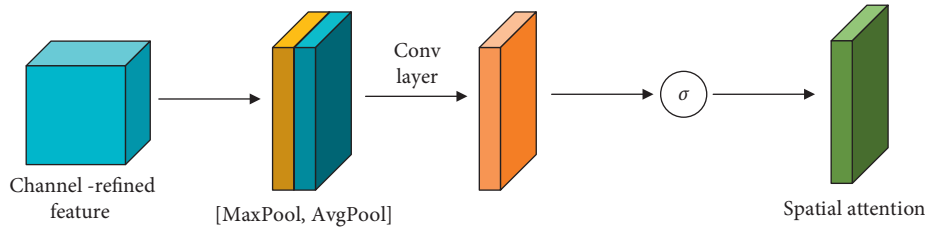


Figure 11: Spatial attention module.

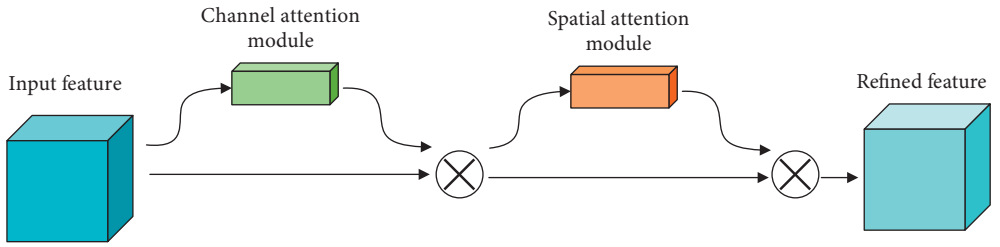


Figure 12: The overview of CBAM.

CBLs in the backbone are replaced with attention modules with no increase in the depth of the network. Figure 13(b) shows our attempt to replace the CBL module in BottleneckCSP (C3) with an attention module, but this proposal was not adopted in the end due to poor results. In addition, the attention module is used to replace CBL modules in the backbone, head, and the entire network, respectively, ultimately choosing to replace only the CBL in the backbone.

2.3.5. Evaluation Indicators in This Article. In this article, the pig detection model uses precision (equation (4)), recall (equation (5)), average precision (equations (6) and (7)), *F*1-score (equation (8)), and mAP as evaluation metrics [35]. The confusion matrix is introduced first in Table 1.

Among them, TP stands for true positive (there is a pig in the image, and the algorithm makes a correct prediction), FP stands for false positive (there is no pig in the image, but the

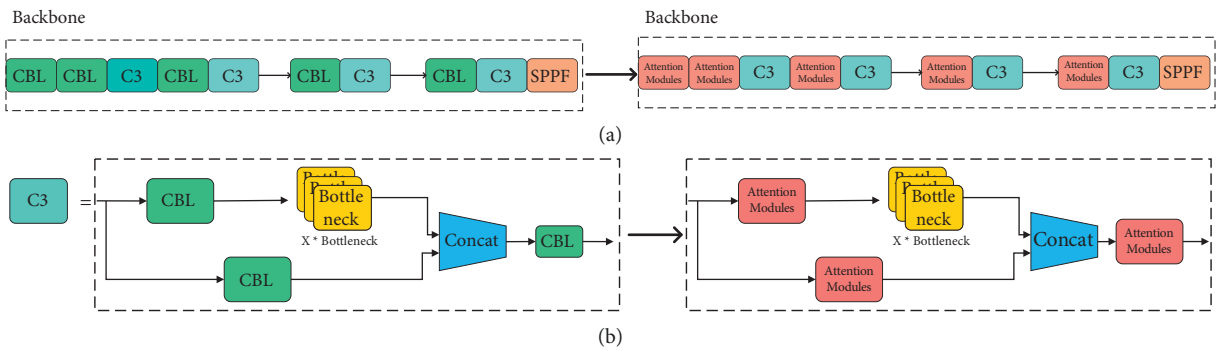


Figure 13: Two methods of combining attention modules with YOLOV5. (a) CBLs in the backbone are replaced by attention modules (our proposed model architecture.). (b) CBLs in BottleneckCSP (C3) are replaced by attention modules.

algorithm detects one), and FN stands for false negative (the algorithm fails to detect the pig in the image). Objects whose IOU with ground truth (e.g., IOU = 0.5) is greater than the threshold are considered TP in the algorithm. Objects with IOU less than the threshold are considered FP; those not correctly identified are considered FN.

Precision represents the proportion of positive samples to those predicted to be positive. The recall represents the proportion of detected positive samples to all positive samples. Average–Precision (AP) refers to the Precision-Recall (PR) curve area. It is relatively difficult to calculate AP with integration, so the interpolation method is commonly used as an alternative. AP@.50 represents the average precision when IOU = 0.5. Similarly, AP@.55 represents the average precision when IOU = 0.55. AP@.50: 95 represents the average of AP@.50, AP@.55, . . ., AP@.95. $F1$-score is the harmonic mean of accuracy and recall. The mAP is the average AP of all classes of objects.

$$\text{precision} = \frac{TP}{TP + FP}, \tag{4}$$

$$\text{recall} = \frac{TP}{TP + FN}, \tag{5}$$

$$AP = \int_0^1 P(R)\mathrm{d}R, \tag{6}$$

$$AP_{50:95} = \frac{1}{10}\left(AP_{50} + AP_{55} + \cdots + AP_{90} + AP_{95}\right), \tag{7}$$

$$F1 - \text{score} = 2 \times \frac{P \times R}{P + R}. \tag{8}$$

3. Results

The model is mainly trained on Google Colaboratory with a Tesla P100 GPU, and the hyperparameters of the model are based on scratch.yaml. Table 2 records some necessary parameter settings of the model.

3.1. Experimental Configuration. In the model's training process, 541 images are used in the training set, and 149 images are used in the validation set. The model parameters are divided into three groups for optimization: weights, bias, and batch-norm. The optimizer uses stochastic gradient descent (SGD) by default. One-cycle learning rate optimization strategy is used during training [36]. In the initial three epochs, the algorithm performs warmup epochs with a larger learning rate (0.01). The learning rate becomes 1/10 of the original one at the end of the warmup epochs. The weight decay ratio is 0.0005. In consideration of the limited memory of the GPU, the batch size is set to 16 in most cases. All algorithms are expected to execute 1000 epochs. However, the algorithm will be stopped immediately if the mAP does not rise within 100 rounds. The formula for mAP calculation here is given in equation (9). To avoid the influence of the pretrained model on the results, the pretrained YOLOV5x model on the MS COCO dataset was not used in the experiments.

Table 1: Confusion matrix.

	Labeled positive	Labeled negative
Predicted positive	TP	FP
Predicted negative	FN	TN

Table 2: The settings of hyperparameters in training.

Parameters	Value
Optimization algorithm	SGD
Epoch	1000
Patience	100
Learning rate	0.01
weight_decay	0.0005
Image size	640
Batch	16 for YOLOV5x and 32 for YOLOV5s
Loss function	BCE loss

$$\text{m}AP = 0.1 \times mAP_{50} + 0.9 \times mAP_{50:95}. \tag{9}$$

3.2. Model Selection. The model selection approach is taken to explore the best solution. A suitable deep learning model is selected from the alternatives. By comparing the performance of different models, we choose the model with the best performance in each metric as the final one.

In this section, we first compare different versions of YOLOV5 and compare the effects of different hyperparameters, including data augmentation and learning rate, on the performance of YOLOV5. The best-performing version of the YOLOV5 model is selected as the baseline model, and it is combined with the attention mechanism to design a new attention mechanism-based YOLOV5 model. We name this model as YOLO_Plus and explore its performance.

First, the four YOLOV5 models were compared. Complex models are generally considered more effective for feature extraction and more susceptible to overfitting. Different versions of YOLOV5 have different depth-multiple and width-multiple to control the width and depth of the network. YOLOV5 has several versions, such as YOLOV5s, YOLOV5l, YOLOV5x, and YOLOV5m. According to the official figures, each model's performance on MS COCO is shown in Table 3.

Table 4 shows the results of YOLOV5s, YOLOV5l, and YOLOV5x on the test images. All models are trained based on scratch.yaml, using SGD as the optimizer and a batch size of 16. It can be seen from the table that YOLOV5x achieves an mAP@.50:.95 of 0.764, outperforming YOLOV5s and YOLOV5l, which are 0.757 and 0.761, respectively. Figure 14 shows the samples of YOLOV5s, YOLOV5l, and YOLOV5x on the same data. All models show good detection performance and can detect most pigs, which confirms the effectiveness of YOLOV5. However, there are still some misses in the detector. For example, in Figure 14(a), YOLOV5s missed two pigs in the top left corner of the image and two pigs in the bottom left corner of the image; in Figure 14(b), one pig in the top left corner of the image was missed by YOLOV5l. Overall, YOLOV5x achieves the best results: it makes fewer mistakes and achieves a high confidence score for the bounding box produced by the same pig. In a nutshell, YOLOV5x can be considered the best model and thus is chosen as the baseline model. Although YOLOV5s have the fastest inference speed of 5.5 ms, their accuracy does not meet the application requirements.

Proper hyperparameter settings are critical to the effectiveness of training. There are more than 30 hyperparameters in YOLOV5 for different training settings, including learning rate, optimizer, loss function, data augmentation, etc. YOLOV5 provides different hyperparameters such as hyp.scratch.yaml, hyp.finetune.yaml, and hyp.finetune_objects365.yaml. hyp.scratch.yaml is the default parameter of YOLOV5 and is optimized for training from scratch on the MS COCO dataset. hyp.finetune.yaml is based on a genetic algorithm evolving 306 generations on the COCO dataset. Table 5 shows the results of the model with different hyperparameters of YOLOV5x. It can be seen from the table that the model trained with hyp.scratch.yaml as hyperparameter outperforms hyp.finetune.yaml in terms of recall, precision, mAP@ .50, and mAP@ .50:.95. These differences are significant. It verifies the great influence of the choice of hyperparameters on the results of the experiments. hyp.scratch.yaml is used for subsequent experiments.

Although YOLOV5x has achieved a high level of precision and recall and can accurately identify pigs, there are still some issues in dealing with specific situations. These problems can be divided into three main categories, as shown in Figure 15. Category A is overlapping, where multiple pigs are not identified due to clustering and stacking. Category B is missing detection mainly caused by the diversity of light conditions and target size. Category C detects backgrounds such as light ropes, feeders, and walls. In general, categories A and C are the most common errors in detection.

According to our conjecture, the primary reason for errors A and B is that the model fails to extract sufficient features with insufficient learning ability. For error C, the main reason is that the model learns mostly color information but not the shape of the pigs. Therefore, it is easy to misidentify the troughs and feeders which share similar colors. It is also challenging to deal with many pigs being stacked together.

Different attention mechanisms have been combined with YOLOV5x to address the above problems, including CBAM, SE, and CoordAtt. Attention mechanisms can emphasize the important information of the object and suppress some irrelevant details, thus better handling the stacking and misidentification cases. Table 6 shows the results of adding different attentions to the model. The CBAM attention module replaces the CBL module in the backbone network. The SE attention is added at the end of the backbone, and the CoordAtt is located in front of the C3 module in the backbone network. As can be seen from the table, in most cases model combined with the attention mechanism can achieve better performance. The best results for all metrics are obtained using YOLOV5x + CBAM. The recall, precision, and mAP@.50:.95 are 0.4%, 0.7%, and 3.2% higher than that of YOLOV5x, respectively.

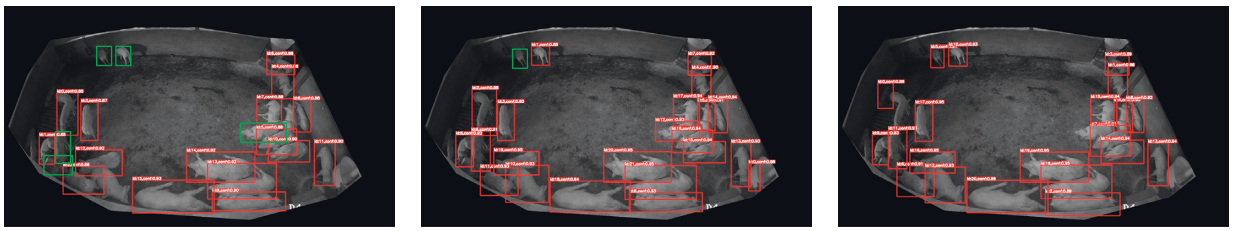
We name the model as YOLOV5_Plus. Figure 16 shows the changes in the relevant metrics during the training process. To better compare the performance of YOLOV5_Plus and YOLOV5x, the samples of detection images obtained using YOLOV5_Plus and YOLOV5x are visualized, as shown in Figure 17. It can be seen that YOLOV5_Plus has made significant improvements in detection capabilities, greatly reducing the number of missed and false detections. The default detection image size by YOLOV5 is 640. However, during the detection process, we found that resizing the image size to 960 can better avoid the false detection of troughs at the edge of the image. Figure 18 shows the sample images after adjusting the image size. In general, the performance of YOLOV5_Plusimg_size = 960 is so impressive. Compared with YOLOV5x, YOLOV5_Plus achieved higher accuracy and recall. By resizing the image to 960 at detection time, the resolution of the image is increased, so it is more effective in detecting overlapping objects and can better avoid false detection of troughs located at the edge of the image. The study attempted to use larger or smaller resolutions for training or detection, but it did not work well. One possible reason is that too large or too small resolutions do not match the preset anchor sizes in YOLOV5x.

Table 3: Performance of YOLOV5s, YOLOV5m, YOLOV5l, and YOLOV5x.

Model	Size (pixels)	mAP(Val) 0.5:0.95	mAP(Val) 0.5	Speed V100 b32 (ms)	Params (M)	FLOPs @640(B)
YOLOV5s	640	37.2	56	0.9	7.2	16.5
YOLOV5m	640	45.2	63.9	1.7	21.2	49
YOLOV5l	640	48.8	67.2	2.7	46.5	109.1
YOLOV5x	640	50.7	68.9	4.8	86.7	205.7

Table 4: Performance metrics for the test images using YOLOV5s, YOLOV5x, and YOLOV5l.

Metric	YOLOV5s	YOLOV5x	YOLOV5l
Recall	0.99	0.992	0.994
Precision	0.981	0.982	0.98
mAP .50	0.993	0.993	0.993
mAP .50:.95	0.757	0.764	0.761
*F*1-score	0.985	0.987	0.987
Inference speed (ms)	5.5	24.1	13



(a) (b) (c)

Figure 14: Examples of pig detection by YOLOV5s, V5l, and V5x with confidence threshold = 0.85. The pigs detected are marked with red boxes, and those not detected or incorrectly detected are marked with green boxes. (a) YOLOV5s. (b) YOLOV5l. (c) YOLOV5x.

Table 5: Performance metrics for the test images using hyp.scratch.yaml and hyp.finetune.yaml.

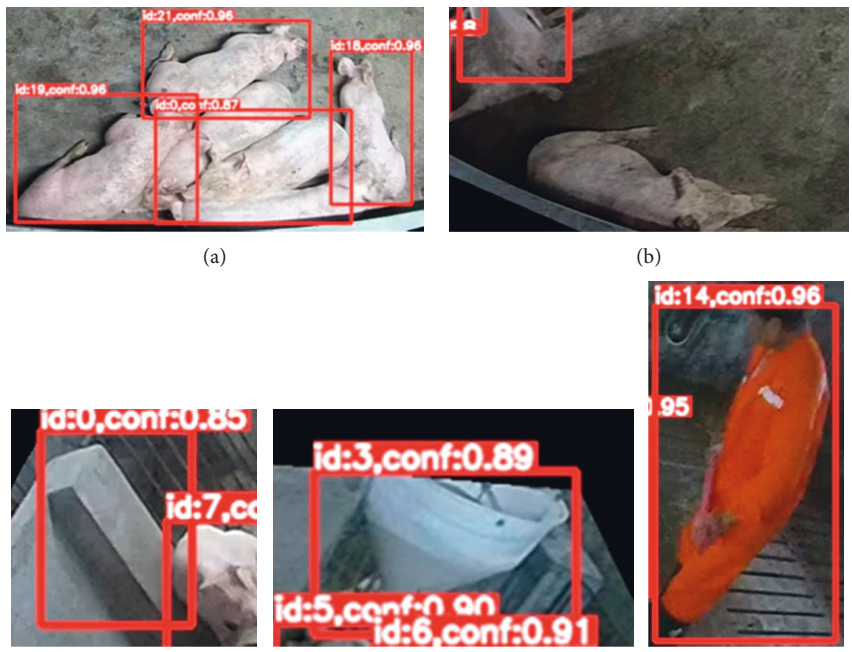
Metric	hyp.scratch.yaml	hyp.finetune.yaml
Recall	0.992	0.984
Precision	0.982	0.973
mAP .50	0.993	0.991
mAP .50:.95	0.764	0.704
F1-score	0.987	0.978
Inference speed	24.1 ms	23.7 ms

However, an interesting phenomenon can be found in Table 6. The combination of YOLOV5x + CoordAtt is shown a decline in performance on almost all metrics. The reason may lie in the potential limitation that the design of the attention mechanism changes with different data structures, and it may be difficult to couple between two different types of attention mechanisms [37].

Table 7 shows the experimental results when the CBAM modules are added to different locations in the YOLOV5x. The attention module replaces the CBL module in the backbone, head, and the whole network of YOLOV5x. The table shows that the best results are obtained when only the CBAM module is applied to the backbone. After adding the attention module to head and backbone + head, the mAP@.50:.95 drops to 0.753 and 0.763, respectively.

We use the CBAM module to replace the CBL module in the backbone and C3 module, respectively, for comparison. The results are shown in Table 8. It can be seen from the table that using the CBAM module to replace the CBL module in the backbone gives better performance and faster inference speed.

Table 9 shows the experimental training results using YOLOV5x pretrained on MS COCO. The results indicate that transfer datasets can significantly improve the performance of the model, with 0.2%, 0.4%, and 3.1% improvement in recall, precision, and mAP@.50: 95, respectively. The results were only slightly lower than those of the untrained YOLOV5_Plus. However, we could not pretrain our model on MS COCO due to limited computing power and time.



(a) (b) (c)

Figure 15: Three typical errors in pig detection by YOLOV5x. (a) Overlap. (b) False negative. (c) False positive.

Table 6: Comparison of different attention modules combined with YOLOV5x.

Algorithms	Recall	Precision	mAP@.50	mAP@.50: 95	*F*1-score	Inference speed (ms)
YOLOV5x	0.992	0.982	0.993	0.764	0.987	24.1
YOLOV5x + CBAM (YOLOV5_Plus)	**0.996**	**0.989**	**0.994**	**0.796**	**0.992**	24.1
YOLOV5x + SE	0.996	0.985	0.994	0.779	0.99	21.6
YOLOV5x + CoordAtt	0.995	0.982	0.994	0.762	0.988	24.9
YOLOV5x + CBAM + SE	0.996	0.988	0.994	0.778	0.992	25.2

The bold values indicate that the best results for all metrics are obtained using YOLOV5x + CBAM.

Other methods, including focal loss and soft-NMS, have been experimented with but have not yielded good results.

3.3. Comparison of Target Detection Algorithms. We compare our YOLOV5_Plus model with YOLOV3 and YOLOV4. Table 10 shows the performance of the above three algorithms. From the table, we can see that YOLOV5_Plus achieves the best results in all metrics. In particular, for the mAP@.50: 95, YOLOV5_Plus achieves a result of 0.796, which improves by 7.1% and 7.9% compared to YOLOV3 and YOLOV4, respectively. These improvements are mainly attributed to innovative modules in YOLOV5, such as Bottleneck_1, Bottleneck_2, Focus, and FPN + PAN for better feature fusion. As a result, YOLOV5 can achieve more efficient information extraction at a very fast speed.

4. Analysis and Discussion

4.1. Findings. Pig detection is an important part of smart animal husbandry. Fast and accurate counting can help farmers improve feeding efficiency and avoid feed waste. However, lighting changes, shooting perspectives, large scale of overlapping, and varying postures of pigs pose many challenges for pig detection. Our results show that using YOLOV5 combined with the attention mechanism can improve accuracy and recall. Overlapping pigs can be better detected, and misidentified backgrounds can be avoided as much as possible.

Before deep learning, counting livestock usually relied on traditional machine learning methods such as RF and SVM. However, deep learning outperformed other methods in most recent cases and showed great potential [38]. Few papers have applied some of the current state-of-the-art detectors to the field. Therefore, our study can be seen as a new attempt at pig detection.

We chose YOLOV5x as the most suitable baseline model for our task in model selection. It outperforms YOLOV5s and YOLOV5l by 0.7% and 0.3%, respectively, in mAP@.50: .95. It is shown that the scale and complexity of YOLOV5x match the task.

Hyperparameters play an essential role in model performance. Inappropriate hyperparameter selection can cause up to 6% degradation in mAP. We suppose that the

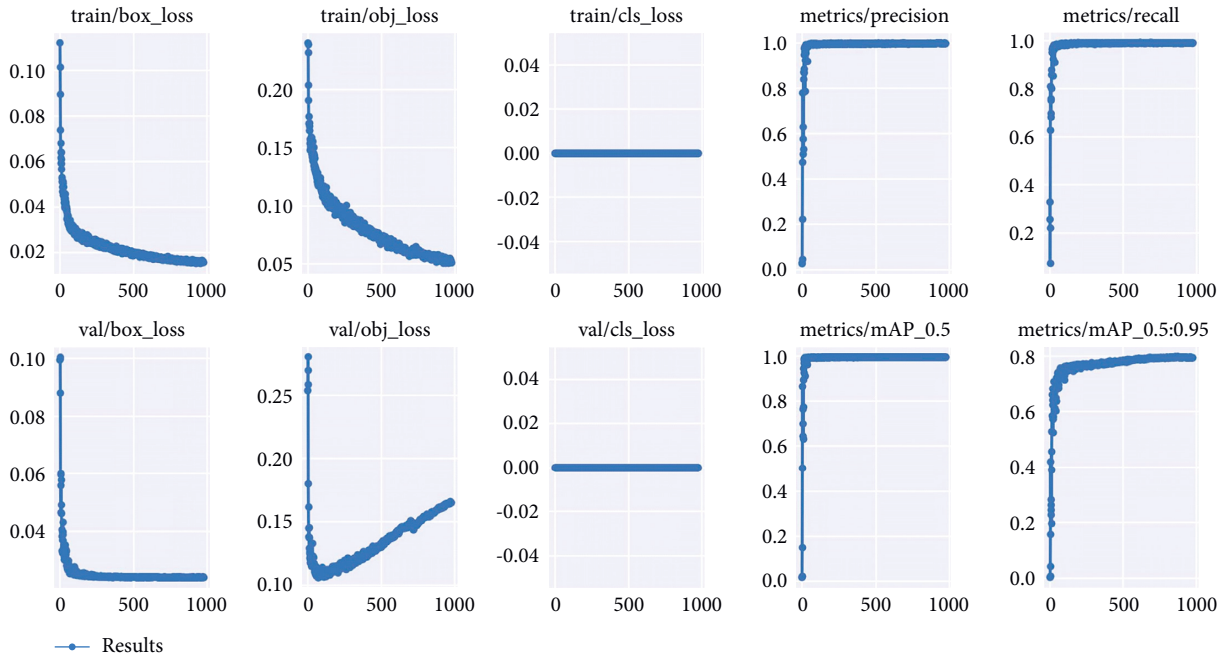


Figure 16: Changes in relevant metrics of training YOLOV5_Plus.

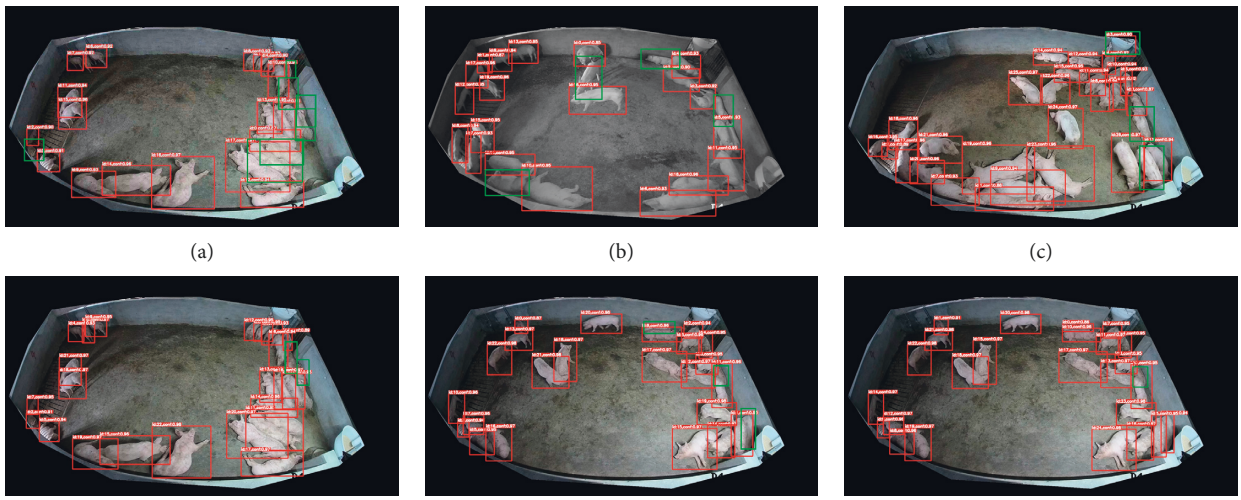


Figure 17: Examples of pig detection by YOLOV5x and YOLOV5_Plus. The first row shows the pictures detected by YOLOV5x. The second row shows the pictures detected by YOLOV5_Plus. The two pictures in the same column are the same. The pigs detected are marked with red boxes, and those not detected or incorrectly detected are marked with green boxes. (a) YOLOV5x_1. (b) YOLOV5x_2. (c) YOLOV5x_3. (d) YOLOV5_Plus_1. (e) YOLOV5_Plus_2. (f) YOLOV5_Plus_3.

degradation of mAP is mainly attributed to the difference in data augmentation. Therefore, appropriate data augmentation is of great importance for the target detection task.

Attention mechanisms can improve the performance of CNNs and YOLO [39]. We tried popular attention mechanisms, including CBAM, SE, and CoordAtt. The results show that most of attention mechanisms can enhance the performance of the model. The experimental results show that the mAP@.50:.95 can be improved up to 3.2% by attention mechanisms.

Pretraining models are widely used in NLP and other AI-related fields. The AI community has become a consensus to use pretrained models instead of training models from scratch [40]. We compared models pretrained on MS COCO with models without pretraining. The results show that pretraining can substantially improve mAP by approximately +3%.

In addition, the inference speed of our model takes only 24.1 ms per image, which means that we can achieve

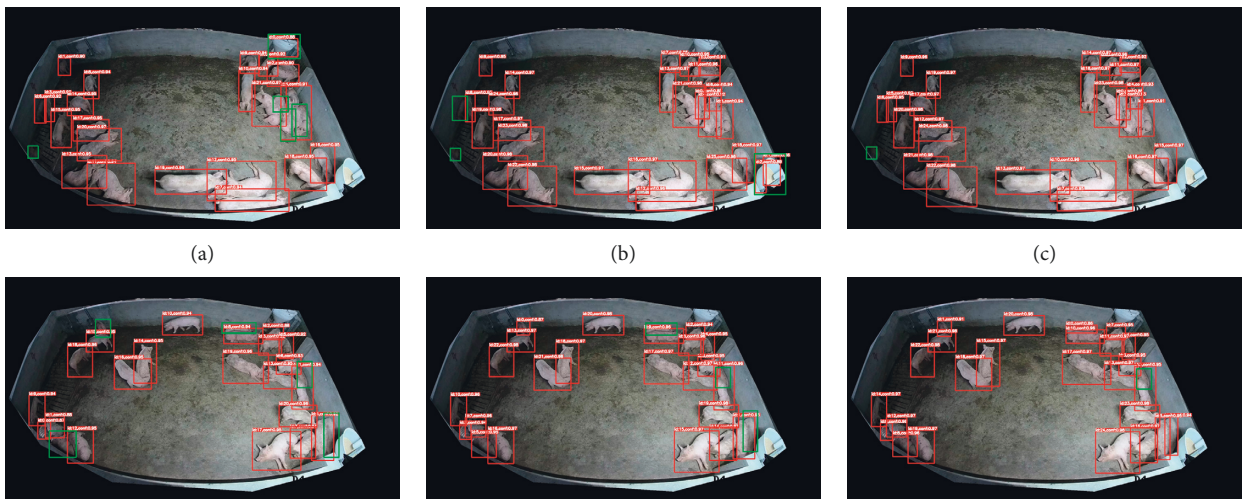


Figure 18: Examples of pig detection by YOLOV5x, YOLOV5_Plus, and YOLOV5_Plus img_size = 960 with a confidence threshold of 0.85. The pigs detected are marked with red boxes, and those not detected or incorrectly detected are marked with green boxes. Top row: compared with YOLOV5x and YOLOV5_Plus, YOLOV5_Plus img_size = 960 has fewer false positive and false negative errors. Bottom row: 20 pigs are detected by YOLOV5x, 22 by YOLOV5_Plus, and 24 by YOLOV5_Plus img_size = 960. (a) YOLOV5x. (b) YOLOV5_Plus. (c) YOLOV5_Plus img_size = 960. (d) YOLOV5x. (e) YOLOV5_Plus. (f) YOLOV5_Plus img_size = 960.

Table 7: Comparison of adding CBAM in different parts of YOLOV5x.

Algorithms	Recall	Precision	mAP@.50	mAP@.50:.95	*F*1-score	Inference speed (ms)
YOLOV5x	0.992	0.982	0.993	0.764	0.987	24.1
YOLOV5x + CBAMbackbone (YOLOV5_Plus)	**0.996**	**0.989**	**0.994**	**0.796**	**0.992**	24.1
YOLOV5x + CBAMhead	0.993	0.98	0.993	0.753	0.986	24.4
YOLOV5x + CBAMbackbone + head	0.996	0.983	0.994	0.763	0.989	24.2

Table 8: Comparison of YOLOV5x with different modules replaced by CBAM.

Algorithms	Recall	Precision	mAP@.50	mAP@.50: 95	F1-score	Inference speed (ms)
YOLOV5x	0.992	0.982	0.993	0.764	0.987	24.1
YOLOV5x + CBAMCBL (YOLOV5_Plus)	**0.996**	**0.989**	**0.994**	**0.796**	**0.992**	24.1
YOLOV5x + CBAMC3	0.994	0.987	0.994	0.791	0.99	26.7

Table 9: Comparison of transfer models using MS COCO transfer dataset with original YOLOV5x.

Algorithms	Recall	Precision	mAP@.50	mAP@.50:.95	*F*1-score	Inference speed (ms)
YOLOV5x	0.992	0.982	0.993	0.764	0.987	24.1
YOLOV5x transfer models	0.994	0.986	0.994	0.795	0.99	23.0

Table 10: Comparison of recall, precision, mAP@.50, and mAP@.50:.95 in methods.

Algorithms	Recall	Precision	mAP@.50	mAP@.50:.95	*F*1-score
YOLOV3	0.971	0.989	0.992	0.725	0.980
YOLOV4	0.988	0.835	0.992	0.717	0.905
YOLOV5x_Plus	0.996	0.989	0.994	0.796	0.992

detection speeds of over 40Fps, perfectly meeting the needs of farm applications.

4.2. Limitations. Our work still has some limitations. First, although our model combined with the attention mechanism has significantly improved in handling overlap situations, it still has difficulty handling test images where types of objects are not presented in the training set, such as feeders and troughs. However, since the objective frames identified by YOLOV5_Plus generally have a higher confidence score, we can screen out most of the troughs by raising the confidence threshold and resizing the test images.

However, there are still some troughs and feeders that cannot be eliminated. We believe this is because the model tends to emphasize the color of the target rather than the shape. The similar posture of pigs and feeders is also one of the possible reasons.

Second, we failed to pretrain our model on MS COCO due to limited computing power and time. The prior knowledge of the pretraining parameters could help our model be trained better and faster. We demonstrated the effectiveness of pretraining using the YOLOV5x model pretrained on MS COCO provided by GitHub.

5. Conclusion

This study presents a deep learning method for pig detection on farms. The YOLOV5_Plus model is a lightweight and efficient model with high accuracy, which is mainly built by combining YOLOV5 and the attention mechanism. The results show that the model can achieve 98.9% recall, 99.6% accuracy, and 79.6% mAP@.50:.95, which are the best results among the competing models. In addition, the size of the detected image can be increased from 640 to 960, which is very effective in avoiding the false detection of troughs. It is worth mentioning that the attention mechanism enables YOLOV5_Plus to improve a lot compared to other algorithms when dealing with dense pigs and falling pigs. Compared with YOLOV3 and YOLOV4, our proposed model achieves higher precision, recall, and mAP. More data augmentation methods will be explored in future research to improve image quality. Meanwhile, detection heads of different scales can be added to meet the difficulties caused by different degrees of overlapping and relatively small images, increasing the generalizability and robustness of the model.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no competing interests.

References

[1] Aga info, https://www.fao.org/ag/againfo/themes/en/pigs/HH_nutrition.html, 2013.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection[C]," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegas NV, June 2016.

[3] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: Single Shot Multibox detector," in *European Conference on Computer Vision*Springer Cham, New York, NY, USA, 2016.

[4] J. Redmon and A. Farhadi, "YOLO9000: better faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, Honolulu, HI, July 2017.

[5] J. Redmon and A. Farhadi, "YOLOV3: An Incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[6] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOV4: Optimal Speed and Accuracy of Object detection," 2020, https://arxiv.org/abs/2004.10934.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, Columbus OH, June 2014.

[8] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, Santiago Chile, December 2015.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 39, no. 6, Montreal, Canada, December 2015.

[10] K. He, G. Gkioxari, P. Dollár, and P. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, Venice, Italy, October 2017.

[11] R. Ranjbarzadeh, S. Jafarzadeh Ghoushchi, M. Bendechache et al., "Lung infection segmentation for COVID-19 pneumonia based on a cascade convolutional network from CT images," *BioMed Research International*, vol. 2021, Article ID 5544742, 16 pages, 2021.

[12] S. Jafarzadeh Ghoushchi, R. Ranjbarzadeh, S. A. Najafabadi, E. Osgooei, and E. B. Tirkolaee, "An extended approach to the diagnosis of tumour location in breast cancer using deep learning," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–11, 2021.

[13] X. Yu, Y. Wang, D. An, and Y. Wei, "Counting method for cultured fishes based on multi-modules and attention mechanism," *Aquacultural Engineering*, vol. 96, Article ID 102215, 2022.

[14] T. Petso, R. S. Jamisola, D. Mpoeleng, E. Bennitt, and W. Mmereki, "Automatic animal identification from drone camera based on point pattern analysis of herd behaviour," *Ecological Informatics*, vol. 66, Article ID 101485, 2021.

[15] P. Ahrendt, T. Gregersen, and H. Karstoft, "Development of a real-time computer vision system for tracking loose-housed pigs," *Computers and Electronics in Agriculture*, vol. 76, no. 2, pp. 169–174, 2011.

[16] M. Tian, H. Guo, H. Chen, Q. Wang, C. Long, and Y Ma, "Automated pig counting using deep learning," *Computers and Electronics in Agriculture*, vol. 163, Article ID 104840, 2019.

[17] M. Riekert, S. Opderbeck, A. Wild, and E. Gallmann, "Model selection for 24/7 pig position and posture detection by 2D camera imaging and deep learning," *Computers and Electronics in Agriculture*, vol. 187, Article ID 106213, 2021.

[18] M. Kashiha, C. Bahr, S. Ott et al., "Automatic identification of marked pigs in a pen using image pattern recognition," *Computers and Electronics in Agriculture*, vol. 93, pp. 111–120, 2013.

[19] L. Wang and W. Q. Yan, "Tree Leaves Detection Based on Deep learning," in *Communications in Computer and Information Science*, pp. 26–38, Springer Cham, New York, NY, USA, 2021.

[20] H. Wang, S. Zhang, S. Zhao, Q. Wang, D. Li, and R. Zhao, "Real-time detection and tracking of fish abnormal behavior based on improved YOLOV5 and SiamRPN++," *Computers and Electronics in Agriculture*, vol. 192, Article ID 106512, 2022.

[21] X. Dong, S. Yan, and C. Duan, "A lightweight vehicles detection network model based on YOLOv5," *Engineering*

Applications of Artificial Intelligence, vol. 113, Article ID 104914, 2022.

[22] AI Studio, "Pig Inventory Challenge," 2021, https://aistudio.baidu.com/aistudio/datasetdetail/96116.

[23] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.

[24] J. Wu, Z. Zhao, C. Sun, R. Yan, and X. Chen, "Reliability engineering and system safety pin Lyu, Hanbin Zhang, Wenbing Yu, Chao Liu. A novel model-independent data augmentation method for fault diagnosis in smart manufacturing," *Procedia CIRP*, vol. 107, pp. 949–954, 2022.

[25] P. Lyu, H. Zhang, W. Yu, and C. Liu, "A novel model-independent data augmentation method for fault diagnosis in smart manufacturing," *Procedia CIRP*, vol. 107, pp. 949–954, 2022.

[26] F. Gao, L. Fu, X. Zhang et al., "Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN," *Computers and Electronics in Agriculture*, vol. 176, Article ID 105634, 2020.

[27] M. Demirtaş, "A new RGB color image encryption scheme based on cross-channel pixel and bit scrambling using chaos," *Optik*, vol. 265, Article ID 169430, 2022.

[28] C. Y. Wang, H. Y. M. Liao, and Y. H. Wu, "CSPNet: A New Backbone that Can Enhance Learning Capability of CNN," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 390-391, Seattle, WA, USA, November 2019.

[29] S. Woo, J. Park, J. Y. Lee, and S. Kweon, "Cbam: Convolutional Block Attention module," *Computer Vision - ECCV 2018*, Springer Cham, New York, NY, USA, 2018.

[30] P. Yan, Q. Sun, N. Yin, L. Hua, S. Shang, and C. Zhang, "Detection of coal and gangue based on improved YOLOv5.1 which embedded scSE module," *Measurement*, vol. 188, Article ID 110530, 2022.

[31] J. Qi, X. Liu, K. Liu et al., "An improved YOLOv5 model based on visual attention mechanism: application to recognition of tomato virus disease," *Computers and Electronics in Agriculture*, vol. 194, Article ID 106780, 2022.

[32] R. Ranjbarzadeh, A. Bagherian Kasgari, S. Jafarzadeh Ghoushchi, S. Anari, M. Naseri, and M. Bendechache, "Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images," *Scientific Reports*, vol. 11, no. 1, Article ID 10930, 2021.

[33] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City UT, June 2018.

[34] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Article ID 13713, Nashville, TN, USA, June 2021.

[35] J. Wang, J. Wei, and S. Mei, "Improved YOLOv3 for small object detection in remote sensing images," *Computer Engineering and Applications*, vol. 57, no. 20, pp. 133–141, 2021.

[36] T. He, Z. Zhang, H. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 558–567, Long Beach CA, June 2019.

[37] J. Kim, S. Lee, E. Hwang et al., "Limitations of deep learning attention mechanisms in clinical research: empirical case study based on the Korean diabetic disease setting," *Journal of Medical Internet Research*, vol. 22, no. 12, Article ID e18418, 2020.

[38] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: a survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.

[39] W. Li, K. Liu, L. Zhang, and F Cheng, "Object detection based on an adaptive attention mechanism," *Scientific Reports*, vol. 10, no. 1, Article ID 11307, 2020.

[40] X. Han, Z. Zhang, and N. Ding, "Pre-trained Models: Past, Present and future," *AI Open*, vol. 2, pp. 225–250, 2021.