agriculture

MDPI

*Article*

# IO-YOLOv5: Improved Pig Detection under Various Illuminations and Heavy Occlusion

**Jiajun Lai [1], Yun Liang [1,2,*], Yingjie Kuang [1], Zhannan Xie [1], Hongyuan He [1], Yuxin Zhuo [1], Zekai Huang [3], Shijie Zhu [1] and Zenghang Huang [3]**

[1] College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China
[2] Guangzhou Key Laboratory of Intelligent Agriculture, South China Agricultural University, Guangzhou 510642, China
[3] College of Engineering, South China Agricultural University, Guangzhou 510642, China
* Correspondence: yliang@scau.edu.cn

**Abstract:** Accurate detection and counting of live pigs are integral to scientific breeding and production in intelligent agriculture. However, existing pig counting methods are challenged by heavy occlusion and varying illumination conditions. To overcome these challenges, we proposed IO-YOLOv5 (Illumination-Occlusion YOLOv5), an improved network that expands on the YOLOv5 framework with three key contributions. Firstly, we introduced the Simple Attention Receptive Field Block (SARFB) module to expand the receptive field and give greater weight to important features at different levels. The Ghost Spatial Pyramid Pooling Fast Cross Stage Partial Connections (GSPPFC) module was also introduced to enhance model feature reuse and information flow. Secondly, we optimized the loss function by using Varifocal Loss to improve the model's learning ability on high-quality and challenging samples. Thirdly, we proposed a public dataset consisting of 1270 images and 15,672 pig labels. Experiments demonstrated that IO-YOLOv5 achieved a mean average precision (mAP) of 90.8% and a precision of 86.4%, surpassing the baseline model by 2.2% and 3.7% respectively. By using a model ensemble and test time augmentation, we further improved the mAP to 92.6%, which is a 4% improvement over the baseline model. Extensive experiments showed that IO-YOLOv5 exhibits excellent performance in pig recognition, particularly under heavy occlusion and various illuminations. These results provide a strong foundation for pig recognition in complex breeding environments.

**Keywords:** object detection; live pig; SARFB; GSPPFC; Varifocal Loss; heavy occlusion; various illumination



**Citation:** Lai, J.; Liang, Y.; Kuang, Y.; Xie, Z.; He, H.; Zhuo, Y.; Huang, Z.; Zhu, S.; Huang, Z. IO-YOLOv5: Improved Pig Detection under Various Illuminations and Heavy Occlusion. *Agriculture* **2023**, *13*, 1349. https://doi.org/10.3390/agriculture13071349

Academic Editor: Andreas Gronauer

Received: 26 May 2023
Revised: 23 June 2023
Accepted: 24 June 2023
Published: 4 July 2023



## 1. Introduction

The pig breeding industry is a crucial component of the agriculture industry as it provides high-quality meat for the food supply chain [1]. With the help of smart agriculture technologies, obtaining precise and real-time statistics on pig populations has become essential. This enables more efficient and data-driven pig breeding practices, allowing for effective distribution of feed and other necessary breeding materials to be implemented.

Traditional animal target detection relies on manual labeling [2] and visual identification that is time-consuming, inefficient, and prone to errors. However, with the rapid development of computer technology, object detection algorithms based on a neural network have been widely implemented in various areas of agriculture [3], including insect pest identification [4], the detection of crop diseases [5], and cattle farming detection [6]. The object detection neural network can be broadly categorized into two-stage and one-stage algorithms. Representative of the two-stage algorithm is the R-CNN [7], while the one-stage algorithm is epitomized by the YOLO [8]. Compared to two-stage algorithms, one-stage algorithms such as the YOLOv5 model have a faster detection speed and fewer background interferences, making them more widely applicable. Within the realm of pig

recognition, Ahn et al. [9] proposed EnsemblePigDet, a pig detection model that utilizes model ensemble techniques to improve accuracy and performance. Sa et al. [10] achieved fast detection of pigs by exploiting complementary information between depth and infrared images. In addition, Huang et al. [11] designed HE-YOLO, a pig recognition model based on multiple attention mechanisms to enhance the object detection performance for pigs. However, it should be noted that experimental datasets for pig detection may not always work in breeding environments. During the breeding process, it is common to encounter issues such as heavy occlusion [12], changes in illumination [13], and so on. These problems can decrease the model's ability to identify pigs effectively.

To improve pig identification under heavy occlusion and varying illuminations, we proposed a new model called IO-YOLOv5 (Illumination-Occlusion YOLOv5) and trained the model using our own collected data which includes various real-world situations. The workflow of IO-YOLOv5 is shown in Figure 1. We utilized the CSPDarknet53 and path aggregation network (PANet [14]) structure of the original YOLOv5 as the backbone and neck. In the backbone of the network, we proposed a more lightweight and effective module named Simple Attention Receptive Field Block (SARFB) inspired by the concept of the Receptive Field Block (RBF [15]), which can enlarge the receptive field of the model and endow different receptive fields with attention domains. Furthermore, the Spatial Pyramid Pooling (SPP) module was optimized into the Ghost Spatial Pyramid Pooling Fast Cross Stage Partial Connections (GSPPFC) module which is based on Ghost Convolution [16] and CSPNet [17], to enhance the reuse of feature information and strengthen the flow of information. To locate pig features under heavy occlusion, we introduced Efficient Attention Channel (ECA [18]) modules to the generated channel attention map. In addition, the Varifocal Loss was utilized as a substitute for the Binary CrossEntropyLoss (BCELoss) during the training phase. This implies that the Varifocal Loss can augment the influence of high-quality and challenging samples on the model while resolving the problem of occlusion. To further improve the model's robustness and generalization ability, we incorporated model ensemble [19] and test time augmentation (TTA). Compared with the baseline model, IO-YOLOv5 provides a lightweight and more accurate model for pig recognition in real breeding environments.

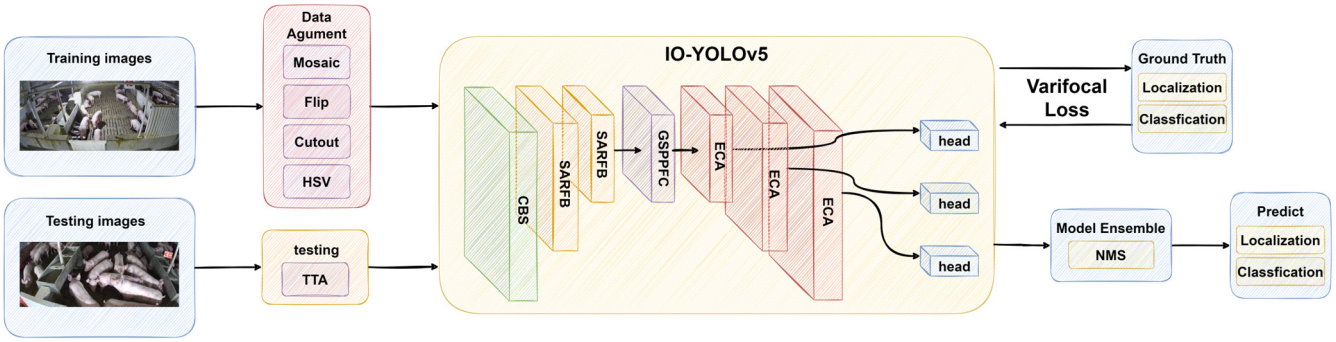



**Figure 1.** Workflow overview for our IO-YOLOv5.

Our main contributions are listed as follows:

- We introduce the GSPPFC module and Simple Attention Receptive Field Block (SARFB) module into YOLOv5 to improve the utilization of features and information transfer, as well as to increase the model's receptive field.
- To the best of our knowledge, this paper presents the first attempt to construct a pig recognition task that involves both various illuminations and heavy occlusions.
- We have gathered and assembled a pig dataset in a real breeding environment with heavy occlusion and various illuminations.

## 2. Materials and Methods

### *2.1. Data Acquisition*

The images used in this study were collected from pig farms located in Yunfu and Dongguan, Guangdong Province, China. Surveillance cameras were placed about 2–3 m above ground level to capture the images. To ensure the high quality of the dataset, images were collected at different times and from different livestock areas. Additionally, we utilized various data augmentation techniques, such as Cutout, Mosaic, brightness adjustment, and random HSV augmentation [20,21], to improve the robustness of our model. This resulted in a total of 1270 RGB images and 15,672 pig labels. On average, each image contains roughly 10–40 pigs. The images were annotated using labeling software labelImg (version 1.8.1), generating individual XML-format files with coordinate markers for each image. These files were then converted into the TXT format required by the YOLO model. Finally, the images and label files were divided into a training–validation set and a testing set at a ratio of approximately 9:1. The training–validation set was further divided into a ratio of approximately 9:1.

In this study, the degree of occlusion between pigs was classified into two categories based on the proportion of the visible area of each pig's characteristics in the image, namely slight occlusion (0–40% occlusion area) and heavy occlusion (over 40% occlusion area). For slight occlusions, as shown in Figure 2a, most pigs were clearly visible with only a few being blocked by other pigs or fences. In contrast, for heavy occlusions, as shown in Figure 2b, the body features of most pigs were obstructed by other pigs or buildings.

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

**Figure 2.** A schematic diagram of the live-pig occlusion: (**a**) The schematic diagram of slight occlusion shown in the picture; (**b**) The picture is a schematic diagram of severe occlusion. The pig selected by the red bounding box in the image is either occluded by the pig selected by the green bounding box or obstructed by other factors.

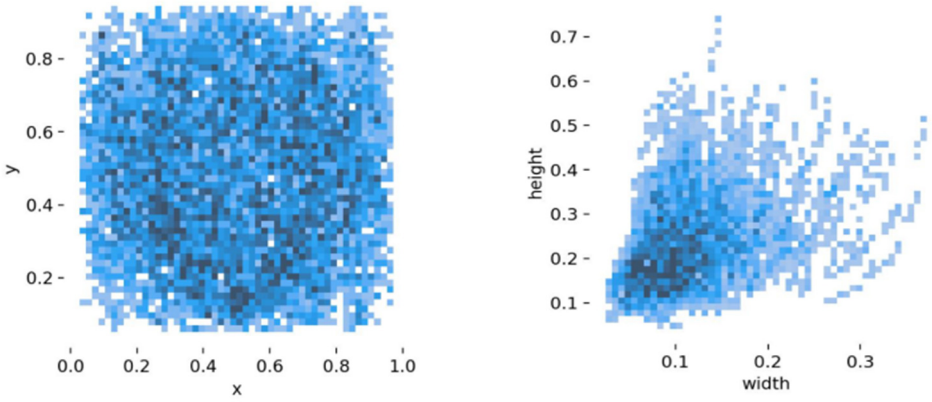
In addition to analyzing different occlusion conditions, we also evaluated the effectiveness of pig detection under different illuminations. We classified illumination conditions into three categories based on their strength: regular light (such as early morning), bright light (such as noon), and darkness (similar to nighttime). Examples for each category are demonstrated in Figure 3.

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

**Figure 3.** A Schematic diagram of the different illumination condition images: (**a**) The figure shows image data under regular illumination, with clear edges; (**b**) The image data under high light irradiation; (**c**) The image data in the dark.

### *2.2. Analysis of the Dataset*

The generated tag file can calculate the center point coordinates of the dataset's labels, as shown in Figure 4a. Based on color depth analysis, the pigs are mostly located in the surround center and the lens edge. Additionally, the length and width of the data tags, as demonstrated in Figure 4b, indicate that the anchor boxes for the pigs, which represent their posture, are mostly vertically oriented.




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

**Figure 4.** Dataset label information; the denser the distribution, the darker the color: (**a**) Distribution of center point. (**b**) Distribution of anchor boxes' height and width.

### *2.3. Baseline Model YOLOv5*

YOLOv5 is an object detection algorithm developed by Ultralytics and it was released in May 2020. It is a one-stage object detection algorithm that can detect multiple types of objects in real-time. Furthermore, YOLOv5 can be divided into four models based on the width and depth of the network: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. Because real-time monitoring of pigs requires efficient models, the smallest YOLOv5s is selected as the basic of model improvement in this paper.

For YOLOv5, it can be mainly divided into two parts: backbone and neck. The backbone is mainly responsible for extracting image features, while the neck is mainly responsible for predicting and outputting features. In YOLOv5, the backbone is mainly constructed using the Cross-Stage Partial Network (CSPNet). This network draws on the idea of skip connections in Resnet [22] but improves upon its shortcomings, resulting in better performance and scalability through the CSP block and the Spatial Pyramid Pooling (SPP [23]) block. The neck is also an important part of the multilevel feature fusion in YOLOv5, mainly composed of the path aggregation network (PANet). Using the PANet module can achieve upsampling of low-level features and downsampling of high-level features, so that information between different stage features can complement each other and work together to achieve better object detection and recognition performance.

### *2.4. Architecture of IO-YOLOv5*

The schematic diagram of IO-YOLOv5 is shown in Figure 5. We optimized YOLOv5 to better adapt to pig recognition tasks with heavy occlusion and various illuminations.

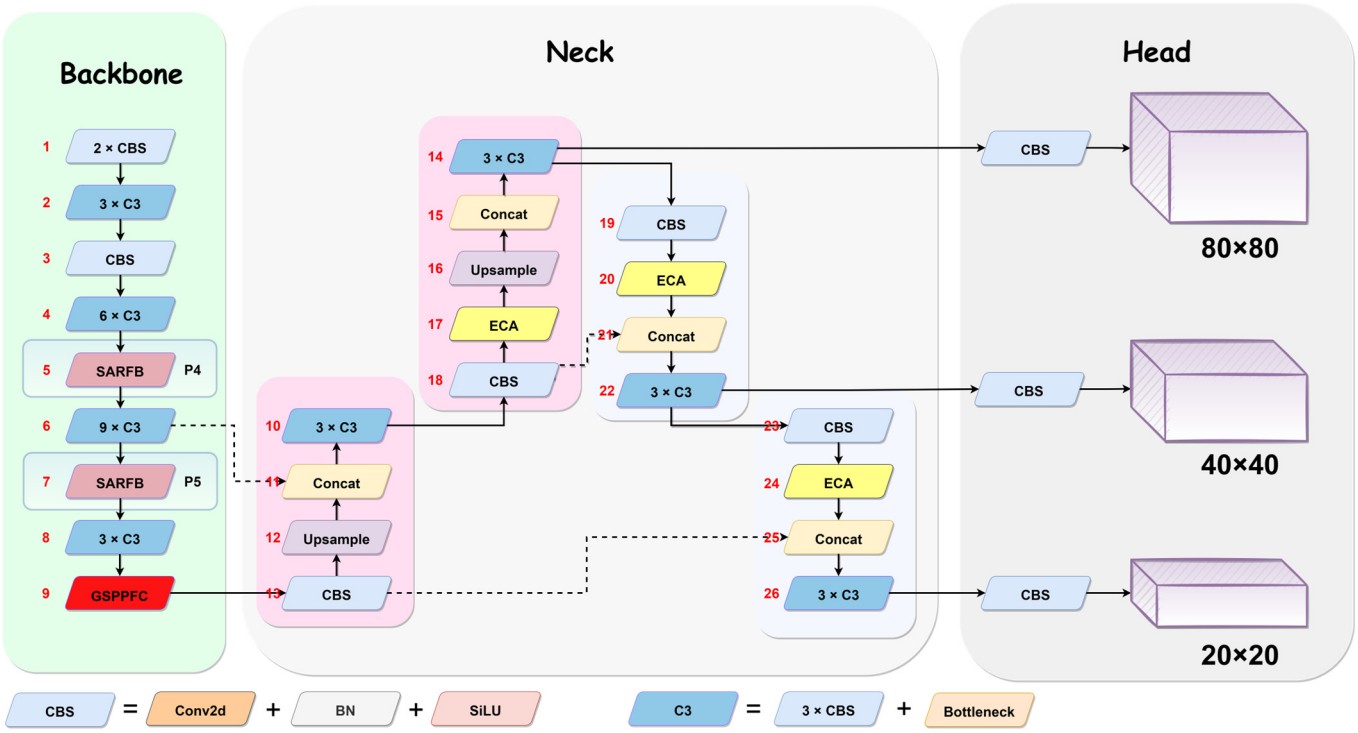



**Figure 5.** The Neural Network Architecture of IO-YOLOv5.

Specifically, we proposed and introduced SARFB modules into the P4 and P5 layers of the network, where SARFB is composed of Conv, Batch Normalization, activation function, and DilatedConv. Additionally, we replaced the SPP module with the GSPPFC module we proposed to enhance feature fusion. The GSPPFC module can maintain the original model's receptive field and the model's inference speed, achieving a lightweight design of the model. The SPP module performs convolution operations on the input feature map using $5 \times 5$, $9 \times 9$, and $13 \times 13$ convolution kernels, followed by Maxpooling for output. In contrast, the GSPPFC module sequentially inputs the input feature map into three $5 \times 5$ convolution kernels, and then merges the results after Maxpooling. The merged output is then fused with the input feature map to preserve the original features.

The training and inference process of the IO-YOLOv5 is presented in Algorithm 1.

#### 2.4.1. Ghost Spatial Pyramid Pooling Fast Cross Stage Partial Connections (GSPPFC)

In YOLOv5, the Spatial Pyramid Pooling (SPP) module is mainly used to merge multiple feature maps and extract spatial features of different scales, enabling the model to acquire diverse receptive fields and enhance its robustness against various objects and scenes. Therefore, we proposes a more effective Ghost Spatial Pyramid Pooling Fast Cross Stage Partial Connections (GSPPFC) module based on the SPP module. First of all, we replace the parallel pooling layers in the original module with cascading pooling layers, to reduce the computational complexity while maintaining similar receptive fields. Then, a CSP module is introduced to establish connections between different network layers, allowing low-level features and high-level features to exchange information and enhancing the model's representation ability. In addition, we introduce a more lightweight convolution module called Ghost Convolution.

**Algorithm 1:** IO-YOLOv5 training and inference process

**Input:** Pig image, target box
**Output:** Predicted box

```
Initialization (learning rate, epochs)
for i in epoch do
    for train_image, target in train_dataloader do
        train_image = augment(train_image)
        output = IO-YOLOv5(train_image)
        loss = VariFocal Loss(output, target)
        loss.backward()
        optimizer.step()
    end
    for val_image, target in val_dataloader do
        output = IO-YOLOv5(val_image)
        metrics(output, target)
    end
    lr_scheduler.step()
    save_model()
end
for test_image in test_dataloader do
    for image in TTA(test_image) do
        output = model_ensemble(IO-YOLOv5(image), YOLOv7-tiny(image))
        merger_output(output)
    end
    save_result(merger_output)
end
```

This module can extract redundant features obtained using some convolutional operations through grouped convolutions, i.e., linear mappings and merge them with key features to obtain the output feature map. Ghost Convolution can obtain redundant features at a smaller computational cost while maintaining important features, with higher computational efficiency and comparable extraction performance. The structure of the Ghost Convolution is shown in Figure 6.

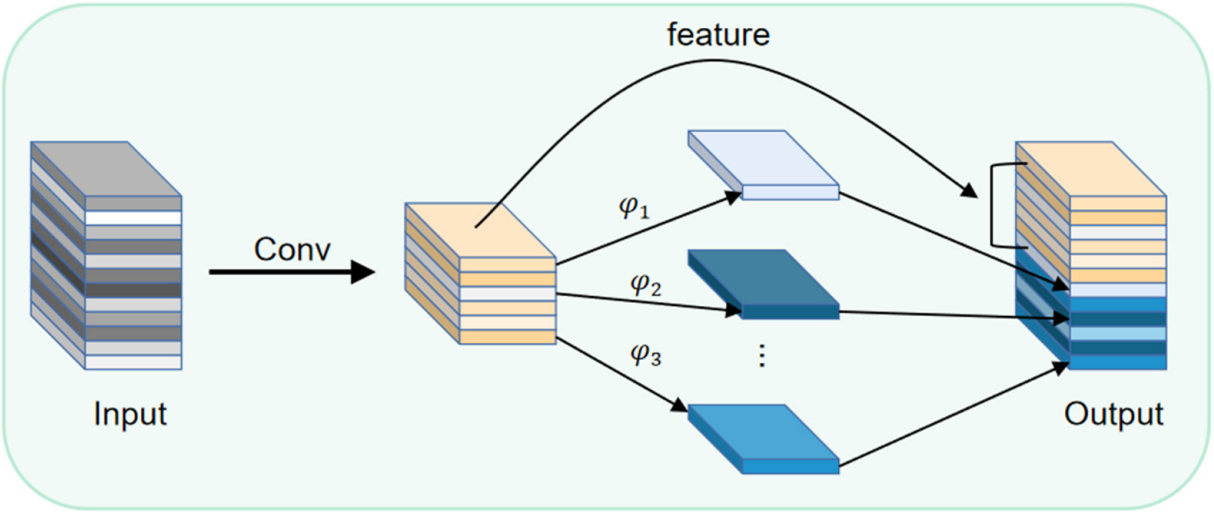



**Figure 6.** The structure of Ghost Convolution module.

For a regular convolutional module and a Ghost Convolution module, assuming their input channel number is $c_{in}$, output channel number is $c_{out}$, and the size of the

convolution kernel is $k \times k$ that is not fixed, the parameter of the regular convolution module ($params_{Conv}$) is

$$params_{Conv} = c_{out} \times k \times k \times c_{in}, \quad (1)$$

due to the fact that the feature map of Ghost Convolution is constructed by concatenating the feature maps of regular convolution and its linear projection. Assuming that the group convolution uses a convolution kernel size of $k\prime \times k\prime$, which was set equal to 5 in the experiment, the parameter of one Ghost Convolution module ($params_{Ghost}$) is

$$params_{Ghost} = \frac{c_{out}}{2} \times k \times k \times c_{in} + k' \times k' \times \frac{c_{out}}{2} \times \frac{c_{out}}{2} \times \frac{2}{c_{out}}. \quad (2)$$

According to Formulas (1) and (2), we can derive that the parameter quantity of Ghost Convolution modules is almost halved compared to that of regular convolutional modules. Similarly, the ratio between floating point operations (FLOPs) is also approximately 1:2.

By combining the Ghost Convolution module and CSP module with the Spatial Pyramid Pooling fast module, this paper proposes a lightweight optimized Spatial Pyramid Pooling module called GSPPFC, whose architecture is shown in Figure 7.

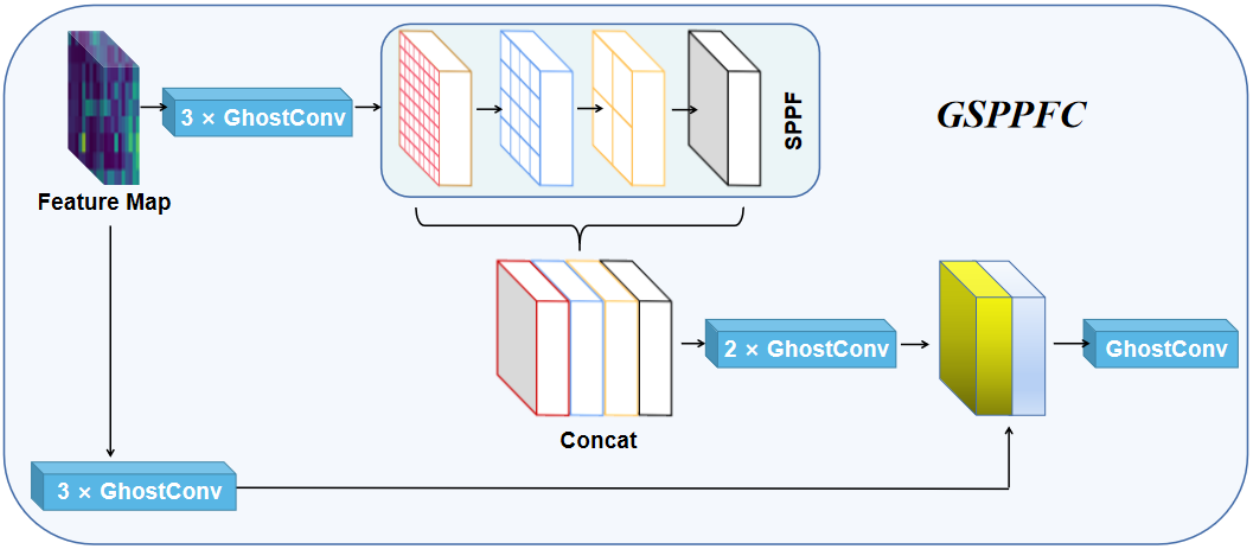



**Figure 7.** The structure of GSPPFC.

Overall, this module replaced the parallel pooling layers with cascading pooling layers and introduced regular convolution with Ghost Convolution on the basis of incorporating the CSP module. On the one hand, this strengthened information exchange between features and, on the other hand, it maintained the lightweight nature of the module.

#### 2.4.2. Simple Attention Receptive Field Block (SARFB)

In the backbone of the network, in addition to optimizing the SPP model, we also proposed a new convolution module inspired by the concept of the Receptive Field Block. We combined regular convolutions and dilated convolutions [24] to extract features and concatenated their feature maps as weights to increase the receptive field of the model. Additionally, we integrated a Simple Attention Module (SimAM [25]) to enhance features under different receptive fields and outputted to the SiLU [26] activation function lastly. By leveraging the environment features surrounding individual pigs, this module can effectively capture the interdependency between features in high-density situations. The schematic diagram of the SARFB module is shown in Figure 8.

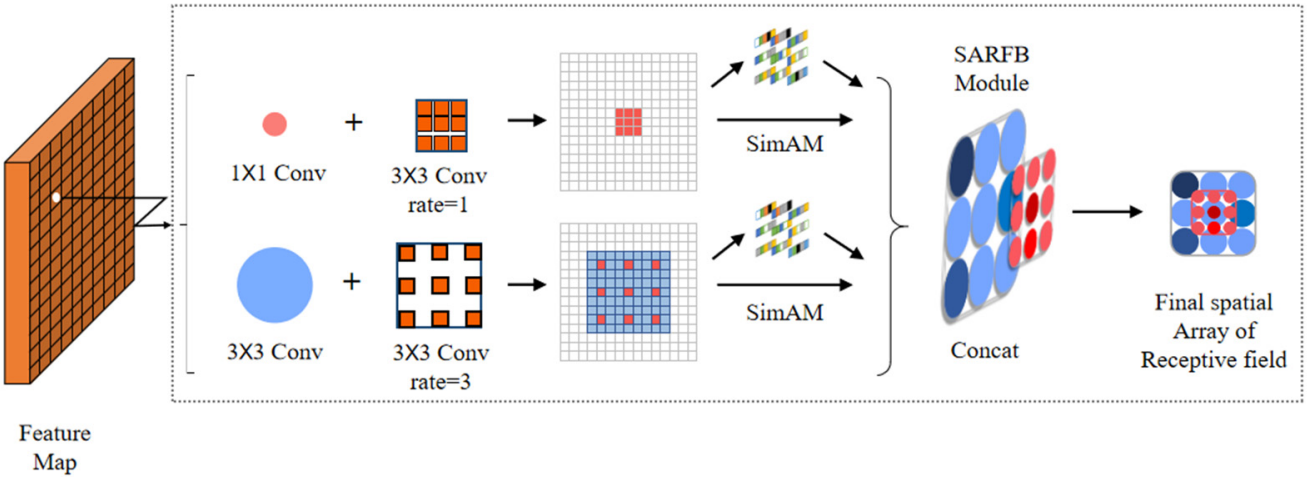



**Figure 8.** Diagram of the SARFB module.

However, increasing the receptive field also lead to feature blurring, which could adversely affect the detection of small objects. To address this issue, we only applied the SARFB module on the two large object detection layers P4 and P5 within the network.

In this module, we utilized a novel Simple Attention Module (SimAM) with a distinctive 3D weight attention mechanism. Compared to other attention mechanisms such as Efficient Channel Attention (ECA) and Squeeze-and-Excitation (SE [27]), SimAM has several advantages, including better performance, and an increased focus on feature weight factors without introducing additional parameters. By employing the SimAM, we can take advantage of the relationships between pixels to capture key features under different receptive fields.

SimAM employs an energy function to assess the significance and interrelation among neurons (i.e., features or pixels) in order to compute the weight. The formula for the energy function can be expressed as

$$e_t(w_t, b_t, y, x_i) = (y_t - \hat{t})^2 + \frac{1}{M-1}\sum_{i=1}^{M-1}(y_o - \hat{x}_i)^2 \quad (3)$$

where $\hat{t} = w_t t + b_t$, $\hat{x}_i = w_t x_i + b_t$, $t$, and $x_i$ are the target neuron and other neurons in a single channel of the input features, respectively. $M$ represents the number of neurons, which is the multiplication of the length and width of the feature maps, while $w_t$ and $b_t$ represent the weight and bias values, respectively.

#### 2.4.3. Efficient Channel Attention (ECA)

In addition to improving the backbone of network, we also made improvements to the network's neck. The Efficient Channel Attention (ECA) module is a simple and effective channel attention that improves upon the Squeeze-and-Excitation (SE) attention mechanism by avoiding the effects of dimensionality reduction on learning the dependencies between channels. The ECA module employs one-dimensional convolution to extract interchannel interactions, enabling local interactions across channels. Moreover, the ECA module is a more lightweight module compared to other attention mechanisms. This module's network structure is illustrated in Figure 9. On a feature map, the ECA module obtains an attention map through channel pooling and fully connected layers, which are then weighted onto the feature map for adaptive feature refinement.

In images with heavy obstruction, the physical features of pigs often get occluded due to overlap between the pigs or obstacles in the buildings. To tackle this issue, the ECA module improves the ability of YOLOv5 to pay more attention to visible pig body features, while also reducing the influence of environmental factors on pig target identification.

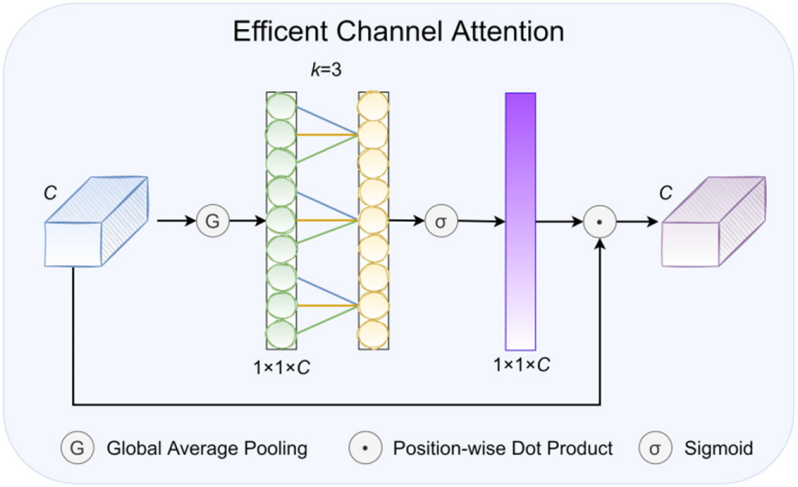



**Figure 9.** The overview of ECA module.

#### 2.4.4. Loss Function Improvement—Varifocal Loss

In addition to enhancing the network model, we have also made significant improvements to the loss function used during the network training process. To calculate the loss in YOLOv5, we usually used the Binary CrossEntropyLoss (BCELoss), which follows the formula shown below:

$$L_n = -w_n[y_n log\sigma(x_n) + (1 - y_n)log(1 - \sigma(x_n))]. \quad (4)$$

In this case, $y$ indicates the ground truth label, $w$ represents the weight parameter, and $x$ represents the predicted result obtained through the Sigmoid activation function.

Based on the Binary CrossEntropyLoss (BCELoss), Varifocal Loss incorporates and optimizes the modulation factor introduced in Focal Loss. By dynamically adjusting the weight of negative samples, the impact of abundant negative samples in prediction is balanced, which, in turn, enhances the model's learning capability for high-quality samples. The calculation formula for VariFocal Loss is shown as follows:

$$VariFocal\ Loss(p, q) = \begin{cases} -q(qlog(p) + (1 - q)log(1 - p)) & q > 0 \\ -\alpha p^{\gamma} log(1 - p) & q = 0 \end{cases} \quad (5)$$

where $p$ is the predicted probability of foreground classes, $q$ is the ground truth, and $\alpha$ and $\gamma$ both are hyperparameters adjusted manually that relate to the influence of negative examples.

In this paper, due to the high-density distribution of pigs, heavy occlusions lead to more negative samples. Varifocal Loss can reduce the impact of these negative examples, effectively improving the model's learning effect.

#### 2.4.5. Model Ensemble and Test Time Augmentation

During the inference phase, we first applied test time augmentation (TTA) to improve the accuracy of predictions. Specifically, TTA consists of the following two steps: (1) Scaling the image to 1.3 times. (2) Horizontally flipping the image. Then, we inputted the three generated images into the model for prediction and used non_max_suppression (NMS) to fuse the results.

Additionally, we performed model ensemble by combining the IO-YOLOv5 and YOLOv7-tiny models [28] that were trained. Similarly, we applied TTA to each model and used NMS to output the fused final prediction results.

## 3. Results

### *3.1. Experimental Environment and Configuration*

The main experimental environment is Python3.8.10, Pytorch1.10.0, CUDA11.3. The specific host configuration is shown in Table 1, and some experimental parameters are shown in Table 2.

**Table 1.** Host configuration.

| Hardware | Specific Configuration |
|---|---|
| Operating System | Ubuntu 9.4.0 |
| CPU | Intel(R) Silver 4310 |
| GPU | RTX A4000 |

**Table 2.** Experimental parameters.

| Experimental Parameters | Specific Parameters |
|---|---|
| Image scaling size | 640 × 640 |
| Number of iterations | 200 |
| batch_size | 8 |
| Optimizer | SGD |
| Initial learning rate | 0.01 |
| Weight decay | 0.0005 |

### *3.2. Evaluation Metrics*

To compare the performance difference between various models and validate the feasibility and effectiveness of proposed improvements, we employ Recall, $F_1$-score, Precision, mean average precision ($mAP_{50}$), frames per second (FPS), and parameters of the model as evaluation metrics to measure a model's performance.

The calculation formulas for the important metrics $AP$ and $mAP$ in object detection are as follows:

$$AP = \int_0^1 p(r)dr, \tag{6}$$

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n}; \tag{7}$$

where p(r) represents the maximum precision at a recall value of $r$, which can be obtained by calculating the highest precision point on each recall segment of the standard PR curve. $n$ represents the total number of categories to be evaluated. The $mAP$ value is usually calculated using Intersection over Union (IoU) with a threshold of 0.5. Therefore, this paper adopts the $mAP_{50}$ when evaluating the performance indicator, which is based on the IoU threshold of 0.5.

### *3.3. Analysis*

#### 3.3.1. Comparison of Different Spatial Pooling Pyramids

To evaluate whether the GSPPFC module used in this experiment performs better than other spatial pooling pyramid modules, we conducted comparative experiments with SPP, SPPF, Atrous Spatial Pyramid Pooling (ASPP [29]), SPPFC, and GSPPC. Additionally, we evaluate the performance using various metrics.

Referring to Table 3, the detection performance of SPP and SPPF is similar, but the detection speed of SPPF is significantly faster than SPP. The SPPFC module that adopts a CSP structure, outperforms the SPP, SPPF, and ASPP modules in two key performance metrics ($F_1$-score and $mAP_{50}$). In terms of the lightweight GSPPFC and GSPPC modules, the former has a lower parameter count than SPPFC, achieves comparable performance in terms of $F_1$-score and $mAP_{50}$ with the original model, and also has a higher FPS score than

all other modules except for SPPF. Moreover, the detection speed and $mAP_{50}$ of GSPPFC are higher.

**Table 3.** Comparison of different space pooling pyramid.

| Module | Precision/% | Recall/% | $F_1$-Score | $mAP_{50}$/% | FPS/f·s$^{-1}$ |
|---|---|---|---|---|---|
| SPP | 88.0 | 79.0 | 0.833 | 88.7 | 85.47 |
| SPPF | 82.7 | 82.2 | 0.825 | 88.6 | 94.34 |
| ASPP | 82.5 | 84.3 | 0.834 | 89.1 | 71.43 |
| SPPFC | 87.0 | 80.9 | 0.838 | 89.5 | 78.74 |
| GSPPC | 86.9 | 81.4 | 0.840 | 89.3 | 81.47 |
| GSPPFC | 84.1 | 83.4 | 0.838 | 89.5 | 86.21 |

3.3.2. Comparison of Different Attention Mechanisms in the Neck

To verify the positive effect of attention mechanisms in the neck on experimental results, and to compare the differences in performance among different attention mechanisms, we conduct a horizontal comparison of four attention mechanisms, including SE, the Convolutional Block Attention Module (CBAM [30]), and ECA. The structure of the SE module is shown in Figure 10.

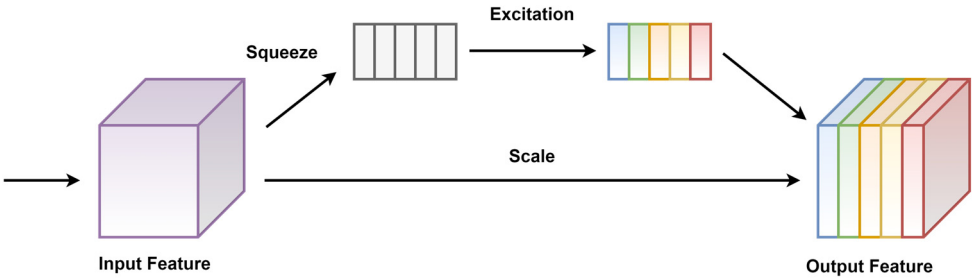



**Figure 10.** The overview of SE module.

The SE module compresses the feature maps into a feature vector, and learns a weight vector for each channel through fully connected layers and activation functions to achieve channel-wise weighting. The structure of CBAM is shown in Figure 11.

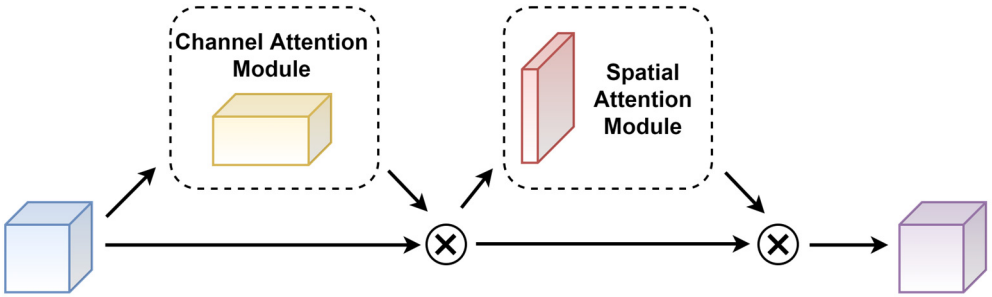



**Figure 11.** The overview of CBAM module.

The CBAM module consists of two submodules, namely the channel attention module and the spatial attention module. Compared to the SE module, CBAM can not only learn the feature information between channels but also learn the attention regions within the feature map.

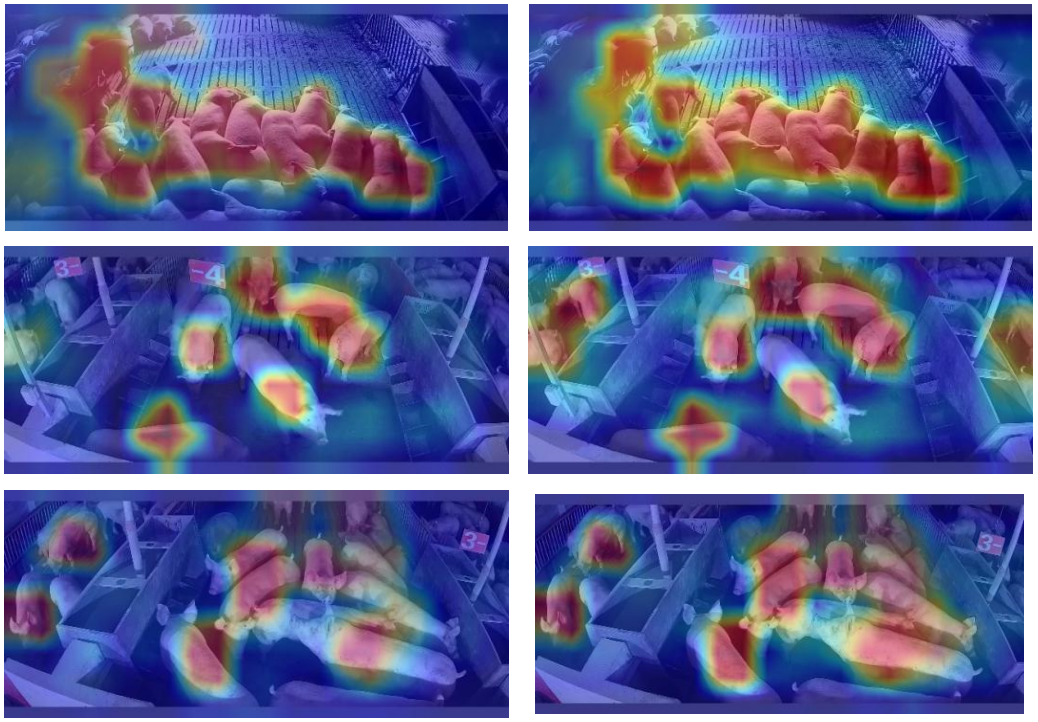
We use various metrics to evaluate these attention mechanisms and determine the best-performing one for use in the neck.

Table 4 shows that the metrics of different attention mechanisms are very similar. After an evaluation of $F_1$-score, $mAP_{50}$, and FPS indicators, we selected ECA as the attention mechanism for the neck in our network due to its outstanding performance. After integrating three ECA modules into the neck, we also perform Class Activation Mapping

(CAM [31]) on both the baseline model and the ECA-YOLOv5 model. The results of CAM are shown in Figure 12. The red highlighted area represents a greater impact on object recognition, while the blue area represents a smaller impact.

**Table 4.** Comparison of different attention mechanisms in the neck.

| Attention | Precision/% | Recall/% | $F_1$-Score | $mAP_{50}$/% | FPS/f·s$^{-1}$ |
|---|---|---|---|---|---|
| SE | 82.8 | 82.3 | 0.826 | 89.1 | 93.46 |
| SimAM | 83.9 | 83.4 | 0.836 | 89.3 | 93.46 |
| CBAM | 83.7 | 82.7 | 0.832 | 89.5 | 91.74 |
| ECA | 82.4 | 84.5 | 0.834 | 89.5 | 95.24 |



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

**Figure 12.** Class Activation Map (CAM) of initial YOLOv5 and YOLOv5 with the ECA module: (**a**) The CAM of the baseline model; (**b**) The CAM of YOLOv5 adding the ECA modules.

It can be seen from Figure 12 that, after incorporating ECA modules into the neck, the model's focus on pig aggregation areas has increased (the color has become brighter and deeper). Additionally, the model also assigns a higher attention weight to areas that were not focused on in the original model but contain pigs.

#### 3.3.3. Ablation Studies of IO-YOLOv5

To verify the impact of different modules on model performance, we conducted ablation experiments on the aforementioned modules for a more comprehensive comparison. "×" denotes the nonuse of a module or method, while "√" indicates the inclusion of a module or method in the model. We compare the results to demonstrate the effectiveness and feasibility of the proposed method and module in improving model performance.

The experiment was based on the baseline model. As shown in Table 5, with the addition of the GSPPFC module, the model's indicators improved overall, including a 0.9% increase in $mAP_{50}$. This proved that the GSPPFC module can effectively improve the backbone's feature extraction ability. By replacing convolution modules in P4 and P5 of the backbone with SARFB modules, the $mAP_{50}$ increased from 89.5% to 90.1%, which also demonstrates that the SARFB module can better extract features of large target pig under

largescale sensitivity fields. After adding the ECA module to the neck of the model and introducing the Varifocal Loss, the final IO-YOLOv5 model was obtained, with an $mAP_{50}$ value of 90.8%, which is 2.2% higher than the baseline model. Finally, after performing model ensemble and using TTA, the $mAP_{50}$ of the model reached 92.6%, which is 1.8% higher than the model without them and 4.0% higher than the baseline model. This ablation experiment effectively demonstrated the effectiveness of the optimization methods and modules proposed in this paper.

**Table 5.** Comparison table of the ablation experiments.

| Strategies | YOLOv5s | Model 1 | Model 2 | Model 3 | Model 4 | Model 5 | Model 6 |
|---|---|---|---|---|---|---|---|
| GSPPFC | × | √ | √ | √ | √ | √ | √ |
| SARFB | × | × | √ | √ | √ | √ | √ |
| Varifocal Loss | × | × | × | √ | √ | √ | √ |
| ECA | × | × | × | × | √ | √ | √ |
| Model Ensemble | × | × | × | × | × | √ | √ |
| TTA | × | × | × | × | × | × | √ |
| Precision/% | 82.7 | 84.1 | 84.7 | 85.7 | 86.4 | 86.3 | 85.1 |
| Recall/% | 82.2 | 83.4 | 83.3 | 81.7 | 82.4 | 87.1 | 87.6 |
| $mAP_{50}$/% | 88.6 | 89.5 | 90.0 | 90.4 | 90.8 | 92.2 | 92.6 |
| FPS/f·s$^{-1}$ | 94.34 | 86.21 | 83.33 | 83.33 | 82.64 | 49.26 | 17.18 |

### *3.4. Algorithm Contrast Experiment*

To further evaluate the performance of the model in pig detection after optimization, we conducted comparison experiments with other network models. Specifically, we compare the optimized model with the Single Shot MultiBox Detector (SSD [32]), YOLOv3, YOLOv4, YOLOv4-tiny, YOLOv7-tiny, and YOLOv7, and the experimental results are shown in Table 6.

**Table 6.** Contrast experiment results of different models.

| Model | Precision/% | Recall/% | $mAP_{50}$/% | Params (M) | FPS/f·s$^{-1}$ |
|---|---|---|---|---|---|
| SSD | 85.4 | 73.6 | 84.1 | 35.007 | 23.26 |
| YOLOv3 | 86.8 | 81.6 | 89.0 | 61.529 | 24.63 |
| YOLOv4 | 81.9 | 72.8 | 89.2 | 64.363 | 31.89 |
| YOLOv4-tiny | 60.3 | 81.2 | 82.6 | 6.057 | 111.40 |
| YOLOv5s | 82.7 | 82.2 | 88.6 | 7.022 | 94.34 |
| YOLOv7-tiny | 84.2 | 80.4 | 87.7 | 6.018 | 101.01 |
| YOLOv7 | 83.9 | 85.5 | 90.4 | 37.202 | 28.74 |
| IO-YOLOv5 | 86.4 | 82.4 | 90.8 | 12.338 | 82.64 |
| IO-YOLOv5 + Ensemble | 86.3 | 87.1 | 92.2 | 18.335 | 49.26 |
| IO-YOLOv5 + Ensemble + TTA | 85.1 | 87.6 | 92.6 | 18.335 | 17.18 |

According to experimental results, our proposed lightweight model IO-YOLOv5 almost perform better than other large network models in all metrics, including SSD, YOLOv3, and YOLOv4. Specifically, for the $mAP_{50}$ metric, IO-YOLOv5 outperforms YOLOv4, which has the best performance among the three, by 1.6%. Additionally, IO-YOLOv5 has a much faster detection speed compared to these three models. Compared to lightweight network models including YOLOv4-tiny, YOLOv5s, and YOLOv7-tiny, IO-YOLOv5 exhibited better accuracy and $mAP_{50}$ performance metrics, with a precision higher than YOLOv7-tiny by 2.2% and $mAP_{50}$ higher than YOLOv5s by 2.2%. Moreover, the FPS of IO-YOLOv5 was more than enough to meet the smooth detection requirements. Finally, after model ensemble with YOLOv7-tiny and using TTA, the FPS of the model decreased, but the detection performance demonstrated significant improvements, making it applicable for scenarios that have different requirements of detection speed and accuracy.

### *3.5. Performance Comparison on the Duck Dataset*

To further validate the effectiveness of our experimental model, we conducted further testing on the Duck dataset [33]. The results are shown in Table 7.

**Table 7.** Contrast experiment results of Duck dataset.

| Method | Precision/% | Recall/% | mAP/% | FPS/f·s$^{-1}$ [1] |
|---|---|---|---|---|
| YOLOv5s | 95.50 | 88.70 | 66.70 | 84.75 |
| YOLOv7 | 95.80 | 93.64 | 65.50 | 30.48 |
| CBAM-YOLOv7 [33] | 96.84 | 94.57 | 66.10 | 29.32 |
| IO-YOLOv5 | 95.23 | 90.26 | 67.17 | 74.62 |

[1] FPS is measured again in the local environment to ensure that it is not affected by hardware conditions. The metric may have some fluctuations.

As shown in Table 7, although IO-YOLOv5 has slightly lower precision and recall than CBAM-YOLOv7, it outperforms CBAM-YOLOv7 in terms of mAP, which is an important indicator. In addition, the detection speed of IO-YOLOv5 is approximately 2.5 times faster than that of CBAM-YOLOv7.

### *3.6. Performance Comparison on Different Pig Dataset*

We also investigated the performance comparison of different models under different task challenges that included various illuminations or heavy occlusion. The result is shown in Table 8.

**Table 8.** Contrast experiment results of Duck dataset.

| Model | Occlusion | Illumination | Precision/% | Recall/% | mAP$_{50}$/% |
|---|---|---|---|---|---|
| [34] | slight | regular | 94.2 | 95.4 | - |
| [10] | slight | various | 92.0 | 86.0 | - |
| [12] | heavy | regular | 90.1 | 92.7 | - |
| IO-YOLOv5 | heavy | various | 85.1 | 87.6 | 92.6 |

As shown in Table 8, complex lighting variations and heavy occlusions can significantly degrade the performance of the models, which is the problem we have explored and addressed in this paper.

## 4. Discussion

### *4.1. Different Occlusion*

In the experiments presented in this paper, we also discuss the effect of live pig recognition under different occlusion. It can be divided into two situations: slight occlusion and heavy occlusion, and the results are shown in Table 9.

**Table 9.** Result of different occlusion.

| Occlusion | YOLOv5s | | | Ours | | |
|---|---|---|---|---|---|---|
| | Precision/% | Recall/% | mAP$_{50}$/% | Precision/% | Recall/% | mAP$_{50}$/% |
| slight | 88.0 | 85.2 | 90.8 | 88.1 | 89.7 | 94.4 |
| heavy | 86.6 | 74.6 | 86.6 | 84.2 | 81.4 | 89.6 |

According to the result in Table 9, the optimized model shows better recognition results than the baseline model under both slight occlusion and heavy occlusion. Specifically, the optimized model achieved a 3.6% and 3.0% increase in mAP$_{50}$ for slight occlusion and heavy occlusion, respectively, compared to the baseline model. This conclusion strongly indicates that our experiment model has superior pig recognition performance under heavy occlusion.

### *4.2. Different Illumination*

We also examined the impact of different illuminations on pig detection. According to the specific breeding environment, it can be divided into three situations: bright, regular, and dark illumination. The specific results are shown in Table 10.

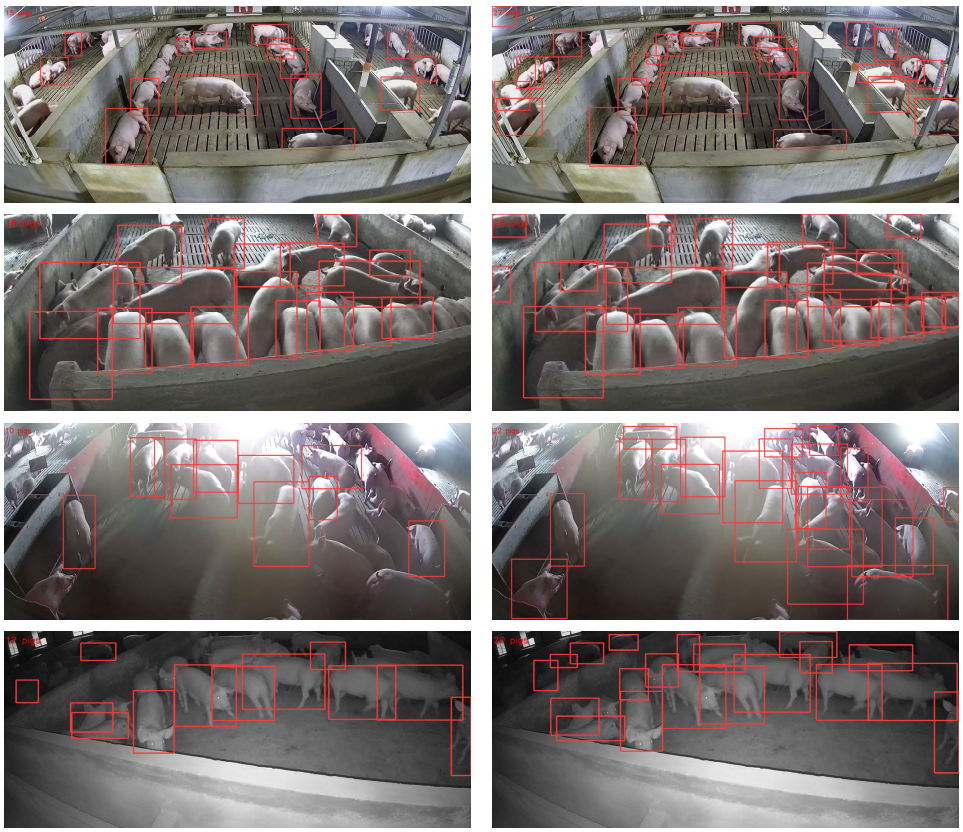
**Table 10.** Result of different illumination.

| Illumination | YOLOv5s | | | Ours | | |
|---|---|---|---|---|---|---|
| | Precision/% | Recall/% | $mAP_{50}$/% | Precision/% | Recall/% | $mAP_{50}$/% |
| bright | 82.8 | 84.5 | 87.2 | 84.3 | 85.2 | 90.8 |
| regular | 86.5 | 83.9 | 90.1 | 86.9 | 88.3 | 94.0 |
| dark | 82.5 | 81.2 | 87.2 | 86.6 | 85.2 | 90.1 |

Based on the result of Table 10, it can be observed that the optimized model performs better in pig recognition than the baseline model under different illumination. Specifically, the optimized model achieved a 3.6%, 3.9%, and 2.9% higher $mAP_{50}$ than the baseline model under bright, normal, and dark illuminations, respectively. This conclusion strongly demonstrates the superior performance of our experiment model in pig detection under various illuminations.

### *4.3. Some Detection Result*

The models we trained allow us to obtain the detection results of some images from YOLOv5 and IO-YOLOv5 on the testing set. Figure 13 shows some pictures of live pig detection results under different illumination and occlusion.



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

**Figure 13.** Some detection results from YOLOv5 and IO-YOLOv5: (**a**) Result of YOLOv5. (**b**) Result of IO-YOLOv5. The red bounding box represents the recognized object, and the number of recognized pigs is annotated in the upper left corner.

The experimental result demonstrates that our model is more capable of identifying pigs in images fairly well across a variety of real-world scenarios than the baseline model.

## 5. Conclusions

This paper proposed a method for recognizing pigs under heavy occlusion and various illuminations based on IO-YOLOv5. The backbone of the network was formed by SARFB modules and a GSPPFC module, and the neck introduced the ECA modules. Meanwhile, the model adopted Varifocal Loss in the training process. The SARFB module consisted of two convolution operations with different kernel sizes and dilation coefficients, which were weighted by SimAM. This allows the model to expand its receptive field and acquire feature highlights from different receptive fields. The GSPPFC used cascade pooling and incorporated the CSP module and Ghost Convolution to enhance the model's information flow. Adding the ECA module enhances the attention domain of features under high occlusion. Ultimately, the IO-YOLOv5 model used Varifocal Loss to improve its ability to learn from high-quality samples.

The experiments showed that IO-YOLOv5 achieved a mAP of 90.8%, which further increased to 92.6% through model ensemble and TTA. Compared to seven other models, IO-YOLOv5 had the highest mAP and its detection speed was three times faster than YOLOv7, the model with the highest mAP among the seven. Furthermore, the mAP of IO-YOLOv5 was 94.4% and 89.6% under slight and heavy occlusion, respectively, and the mAP was 90.8%, 94.0%, and 90.1% under bright, regular, and dark illuminations, respectively, all of which were higher than the baseline model.

IO-YOLOv5 improves the accuracy of pig recognition under heavy occlusion and various illuminations, indicating that the model has good performance. However, compared to the baseline model, the detection speed of this model has decreased somewhat, but it still maintains a high level of 82.64 $f \cdot s^{-1}$ and can meet the detection needs of most production environments.

This experiment has optimized the accuracy of pig recognition under high occlusion and different lighting conditions by expanding the model's receptive field, enhancing the model's feature fusion, and strengthening the model's learning of high-quality samples. In addition, we can further explore the possibility of model optimization by using covariance pooling to introduce second-order features and finding differences in channel features. This is also the direction we hope to continue exploring in our next step.

**Author Contributions:** Conceptualization, J.L. and Y.L.; methodology, J.L.; software, Y.Z. and S.Z.; validation, J.L. and Z.X.; formal analysis, J.L., Z.X. and H.H.; investigation, Z.X., H.H. and Z.H. (Zenghang Huang); resources, Y.K.; data curation, J.L., Z.X. and S.Z.; writing—original draft preparation, J.L.; writing—review and editing, J.L. and Y.L.; visualization, H.H.; supervision, J.L.; project administration, Y.L.; funding acquisition, Y.L. and Z.H. (Zekai Huang). All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National College Student Innovation and Entrepreneurship Training Program Support Project (202210564066), the National Natural Science Foundation of China (61772209), and the key R&D project of Guangzhou (202206010091, 2023B03J1363), Special Fund for Rural Revitalization Strategy of Guangdong (2023TS-3).

**Institutional Review Board Statement:** The animal study protocol was approved by the Institutional Review Board (or Ethics Committee) of the Guangdong Provincial Laboratory Animal Welfare and the Ethical Review Guidelines and were approved by the Animal Welfare Committee of South China Agricultural University (No: 2021F129).

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to restrictions, e.g., privacy or ethical.

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

## References


1. Mugonya, J.; Kalule, S.W.; Ndyomugyenyi, E.K. Effect of Market Information Quality, Sharing and Utilisation on the Innovation Behaviour of Smallholder Pig Producers. *Cogent Food Agric.* **2021**, *7*, 1948726. [CrossRef]
2. Brünger, J.; Traulsen, I.; Koch, R. Model-Based Detection of Pigs in Images under Sub-Optimal Conditions. *Comput. Electron. Agric.* **2018**, *152*, 59–63. [CrossRef]
3. 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]
4. Kasinathan, T.; Singaraju, D.; Uyyala, S.R. Insect Classification and Detection in Field Crops Using Modern Machine Learning Techniques. *Inf. Process. Agric.* **2021**, *8*, 446–457. [CrossRef]
5. Kendler, S.; Aharoni, R.; Young, S.; Sela, H.; Kis-Papo, T.; Fahima, T.; Fishbain, B. Detection of Crop Diseases Using Enhanced Variability Imagery Data and Convolutional Neural Networks. *Comput. Electron. Agric.* **2022**, *193*, 106732. [CrossRef]
6. Qiao, Y.; Guo, Y.; He, D. Cattle Body Detection Based on YOLOv5-ASFF for Precision Livestock Farming. *Comput. Electron. Agric.* **2023**, *204*, 107579. [CrossRef]
7. 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, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
8. 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, 27–30 June 2016; pp. 779–788.
9. Ahn, H.; Son, S.; Kim, H.; Lee, S.; Chung, Y.; Park, D. EnsemblePigDet: Ensemble Deep Learning for Accurate Pig Detection. *Appl. Sci.* **2021**, *11*, 5577. [CrossRef]
10. 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]
11. Huang, L.; Xu, L.; Wang, Y.; Peng, Y.; Zou, Z.; Huang, P. Efficient Detection Method of Pig-Posture Behavior Based on Multiple Attention Mechanism. *Comput. Intell. Neurosci.* **2022**, *2022*, 1–12. [CrossRef] [PubMed]
12. A Light-Weight and Accurate Pig Detection Method Based on Complex Scenes. Available online: https://link.springer.com/article/10.1007/s11042-022-13771-6 (accessed on 23 May 2023).
13. Psota, E.T.; Mittek, M.; Pérez, L.C.; Schmidt, T.; Mote, B. Multi-Pig Part Detection and Association with a Fully-Convolutional Network. *Sensors* **2019**, *19*, 852. [CrossRef] [PubMed]
14. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
15. Liu, S.; Huang, D.; Wang, Y. Receptive Field Block Net for Accurate and Fast Object Detection. In Proceedings of the Computer Vision–ECCV 2018, Munich, Germany, 8–14 September 2018; pp. 404–419.
16. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1577–1586.
17. Wang, C.-Y.; Liao, H.-Y.M.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A New Backbone that Can Enhance Learning Capability of CNN. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 13–19 June 2020; pp. 1571–1580.
18. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11531–11539.
19. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, QC, Canada, 10–17 October 2021; pp. 2778–2788.
20. Hernández-Hernández, J.L.; García-Mateos, G.; González-Esquiva, J.M.; Escarabajal-Henarejos, D.; Ruiz-Canales, A.; Molina-Martínez, J.M. Optimal Color Space Selection Method for Plant/Soil Segmentation in Agriculture. *Comput. Electron. Agric.* **2016**, *122*, 124–132. [CrossRef]
21. García-Mateos, G.; Hernández-Hernández, J.L.; Escarabajal-Henarejos, D.; Jaén-Terrones, S.; Molina-Martínez, J.M. Study and Comparison of Color Models for Automatic Image Analysis in Irrigation Management Applications. *Agric. Water Manag.* **2015**, *151*, 158–166. [CrossRef]
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In Proceedings of the Computer Vision–ECCV 2014, Zurich, Switzerland, 6–12 September 2014; pp. 346–361.
24. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2016**, arXiv:1511.07122.
25. Yang, L.; Zhang, R.-Y.; Li, L.; Xie, X. SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. In Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 11863–11874.
26. Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. *Neural Netw.* **2018**, *107*, 3–11. [CrossRef] [PubMed]

27. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
28. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv* **2022**, arXiv:2207.02696.
29. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
30. CBAM: Convolutional Block Attention Module. Available online: https://link.springer.com/chapter/10.1007/978-3-030-01234-2%20_1 (accessed on 8 May 2023).
31. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. [CrossRef]
32. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision–ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
33. Jiang, K.; Xie, T.; Yan, R.; Wen, X.; Li, D.; Jiang, H.; Jiang, N.; Feng, L.; Duan, X.; Wang, J. An Attention Mechanism-Improved YOLOv7 Object Detection Algorithm for Hemp Duck Count Estimation. *Agriculture* **2022**, *12*, 1659. [CrossRef]
34. Wutke, M.; Heinrich, F.; Das, P.P.; Lange, A.; Gentz, M.; Traulsen, I.; Warns, F.K.; Schmitt, A.O.; Gültas, M. Detecting Animal Contacts—A Deep Learning-Based Pig Detection and Tracking Approach for the Quantification of Social Contacts. *Sensors* **2021**, *21*, 7512. [CrossRef] [PubMed]