Novel multiple object tracking method for yellow feather broilers in a flat breeding chamber based on improved YOLOv3 and deep SORT

Xiuguo Zou[1*], Zhengling Yin[1], Yuhua Li[1], Fei Gong[1], Yungang Bai[2], Zhonghao Zhao[1], Wentian Zhang[3], Yan Qian[1], Maohua Xiao[2]

(1. *College of Artificial Intelligence, Nanjing Agricultural University, Nanjing 210095, China*;
2. *College of Engineering, Nanjing Agricultural University, Nanjing 210095, China*;
3. *Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia*)

Abstract: Aiming at the difficulties of the health status recognition of yellow feather broilers in large-scale broiler farms and the low recognition rate of current models, a novel method based on machine vision to achieve precise tracking of multiple broilers was proposed in this paper. Broilers' behavior in the breeding environment can be tracked to analyze their behaviors and health status further. An improved YOLOv3 (You Only Look Once v3) algorithm was used as the detector of the Deep SORT (Simple Online and Realtime Tracking) algorithm to realize the multiple object tracking of yellow feather broilers in the flat breeding chamber, which replaced the backbone of YOLOv3 with MobileNetV2 to improve the inference speed of the detection module. The DRSN (Deep Residual Shrinkage Network) was integrated with MobileNetV2 to enhance the feature extraction capability of the network. Moreover, in view of the slight change in the individual size of the yellow feather broiler, the feature fusion network was also redesigned by combining it with the attention mechanism to enable the adaptive learning of the objects' multi-scale features. Compared with traditional YOLOv3, improved YOLOv3 achieves 93.2% mAP (mean Average Precision) and 29 fps (frames per second), representing high-precision real-time detection performance. Furthermore, while the MOTA (Multiple Object Tracking Accuracy) increases from 51% to 54%, the IDSW (Identity Switch) decreases by 62.2% compared with traditional YOLOv3-based objective detectors. The proposed algorithm can provide a technical reference for analyzing the behavioral perception and health status of broilers in the flat breeding environment.

Keywords: yellow feather broiler, flat breeding chamber, multiple object tracking, improved YOLOv3, Deep SORT
DOI: 10.25165/j.ijabe.20231605.7836

Citation: Zou X G, Yin Z L, Li Y H, Gong F, Bai Y G, Zhao Z H, et al. Novel multiple object tracking method for yellow feather broilers in a flat breeding chamber based on improved YOLOv3 and deep SORT. Int J Agric & Biol Eng, 2023; 16(5): 44–55.

1 Introduction

Growing global demand for broilers requires large-scale breeding[1]. Yellow feather broiler enjoys a reputation for its fast growth, robust characteristics, and delicacy meat. The consumption of yellow feather broilers in 2020 reached 4.5 billion, which is equivalent to that of white feather broilers. In recent years, the continuous expansion of yellow feather broiler breeding has posed more challenges for refined management. Real-time perception of yellow feather broilers' behaviors and their movement status will help detect abnormalities in time[2], thus improving broiler quality and yield. Multiple object tracking is significant to the yellow feather broiler breeding industry as the basis of real-time behavior perception.

Multiple object tracking is more complicated than single object tracking. In multiple object tracking, in addition to object deformation and background interference, the following problems need to be solved: (1) the appearance of new objects and the disappearance of old objects, (2) object re-identification, that is, accurately distinguishing each object, (3) the interaction and occlusion between objects, (4) the re-recognition of the disappearing object when it appears again. The complex scenarios of livestock breeding, unstable lighting conditions, the gathering behavior of livestock, and the similarity between the livestock of the same species make it difficult to track multiple objects accurately and attract considerable attention in the application of multi-objective tracking in animal research areas. Fujii et al.[3] developed a poultry tracking system based on a particle filtering algorithm for analyzing the behavior of poultry infected with avian influenza. Based on the support maps pointing to preliminary pig segments, Ahrendt et al.[4] built a 5D-Gaussian model of the individual pigs for the real-time tracking of pigs in loose-housed stables. Nakarmi et al.[5] used 3D computer vision and RFID (Radio Frequency Identification) technologies to develop an automated tracking and behavior quantification system for individual broilers housed in groups. Mittek et al.[6] presented a system that used depth images to track individual pigs in a group-housed environment. The tracking method used by this system applied expectation maximization as a policy for fitting an ellipsoid to each pig.

Received date: 2022-08-03 **Accepted date:** 2023-02-19
Biographies: Zhengling Yin, Bachelor, research interest: poultry phenotype, Email: 32318123@njau.edu.cn; **Yuhua Li**, PhD, Lecturer, research interest: poultry phenotype and plant phenotype, Email: lyhresearch@njau.edu.cn; **Fei Gong**, Bachelor, research interest: poultry phenotype, Email: 32219405@njau.edu.cn; **Yungang Bai**, Master, research interest: poultry phenotype, Email: 2020112043@stu.njau.edu.cn; **Zhonghao Zhao**, Bachelor, research interest: poultry phenotype, Email: 9183011329@njau.edu.cn; **Wentian Zhang**, PhD, Research Associate, research interest: machine learning, deep learning, and signal processing, Email: wentian.zhang@alumni.uts.edu.au; **Yan Qian**, PhD, Associate Professor, research interest: poultry phenotype, Email: qianyan@njau.edu.cn; **Maohua Xiao**, PhD, Professor, research interest: intelligent control of agricultural machinery equipment, Email: xiaomaohua@njau.edu.cn.
***Corresponding author: Xiuguo Zou**, PhD, Associate Professor, research interest: poultry phenotype and inspection robot. College of Artificial Intelligence, Nanjing Agricultural University, No.1 Weigang, Nanjing 210095, China. Tel: +86-25-58606585, Email: zouxiuguo@njau.edu.cn.

The tracking effect of the multiple object tracking algorithm mainly depends on the accuracy of object detection. The performance of object detection algorithms improves significantly with the rapid development of deep learning. The deep learning-based object detection algorithms can be divided into one-stage detection algorithms and two-stage detection algorithms. One-stage algorithms detect objects in images using a single deep neural network. In contrast, two-stage algorithms propose a set of regions of interest by selecting a search or regional proposal network first and then using a classifier to process the region candidates. One-stage algorithms mainly include YOLO (You Only Look Once) algorithms[7-9], SSD (Single Shot MultiBox Detector)[10-12], RetinaNet[13], etc. Qin et al.[14] proposed a robust online multiple pig detection and tracking method. This method used SSD to achieve detection, a correlation filtering algorithm to achieve tracking, and a novel hierarchical data association algorithm to match detection and tracking results. Sun et al.[15] used the foreground detection method based on color features and the YOLOv3 algorithm to quickly identify the yellow feather broilers and then used SORT (Simple Online and Realtime Tracking) to track the yellow feather broilers in a flat breeding chamber. Compared with the foreground detection method, the precision and recall of the YOLOv3 algorithm were improved by 11.3% and 17.5%, respectively. Two-stage algorithms mainly include R-CNN (Region-based Convolutional Neural Network)[16], Fast R-CNN[17], Faster R-CNN[18], etc. Based on Faster R-CNN, Sun et al.[19] solved the problem of target tracking frame loss in the visual tracking of pigs. Lin et al.[20] used Faster R-CNN to detect and track broilers. The activity level of broilers was calculated from the tracking results and combined with the THI (Temperature and Humidity Index) value as a new predictive index to avoid heat stress in broilers. However, the methods proposed above tend to run slowly in practical applications. To solve this issue, the backbone of the YOLOv3 detector was replaced with the lightweight MobileNetV2, aiming to reduce computational load and thereby accelerate the model's inference speed. Additionally, considering the need to perform multi-object tracking in the complex scene, attention mechanisms were introduced to enhance the network's feature extraction capability.

The accuracy of the tracking results and the real-time ability are important for the multiple objective tracking algorithms. In terms of object detection, the one-stage detection algorithm has a faster inference speed than the two-stage detection algorithm, which matches the requirements of multiple object tracking. Among the one-stage detection algorithms, the detection accuracy of the YOLOv3 algorithm is higher than that of most other algorithms[9]. In terms of multiple object tracking algorithms, tracking algorithms based on the Hungarian algorithm, such as SORT[21] and Deep SORT[22], can meet the requirements of real-time tracking. Moreover, as an improvement of SORT, Deep SORT extracts the appearance information of targets through a small CNN. Therefore, Deep SORT realizes the re-tracking after the target disappears for a short time, improving the tracking effect of multiple objects.

The combination of improved YOLOv3 and Deep SORT-based tracking algorithm of yellow feather broilers was proposed in this paper. The proposed algorithm can also be modified, considering the size of the yellow feather broiler in the flat breeding chamber does not change much, and they often aggregate together. This research improved the detection accuracy of YOLOv3 by introducing the DRSN (Deep Residual Shrinkage Network), modifying the feature fusion network, and using the attention mechanism to strengthen the target feature. Moreover, by replacing the backbone of YOLOv3, the complexity of the network was simplified, and the detection speed was improved. As a result, the tracking effect and tracking speed of the entire algorithm were improved. This method could achieve real-time and accurate tracking of the broilers' movement, providing data support for analyzing the health status of broilers.

2 Materials and methods

2.1 Data acquisition

The broiler house used in this experiment was built in Jinniuhu Subdistrict, Luhe District, Nanjing City, Jiangsu Province, China. The broiler house has two chambers with symmetrical structures. Each chamber has a width of 1.9 m, a length of 2.9 m, and a total area of 5.51 m^2[23]. Forty-five yellow feather broilers were raised in each chamber. The video surveillance system comprised HIKVISION's CS-C4W-3C2WFR dome network camera, monitoring host, and network hard disk video recorder. The camera has a focal length of 2.8 mm and a pixel of 2 million.

2.2 Overall technical route

This paper used a deep learning-based multiple object tracking algorithm to track the yellow feather broilers in the flat breeding chamber. The technical route of this experiment (shown in Figure 1) includes four main parts:

(1) Construction of the object detection dataset and multiple object tracking dataset.

(2) Training of the object detection module. Train the improved YOLOv3 algorithm with the object detection dataset and use it as a detector of multiple object tracking.

(3) Realization of multiple object tracking of yellow feather broilers. Combine the improved YOLOv3 and Deep SORT to realize the tracking of yellow feather broilers.

(4) Result analysis. Evaluate the proposed tracking algorithm based on some key performance parameters (e.g., mAP (mean Average Precision), FPS (Frames Per Second), MOTA (Multiple Object Tracking Accuracy), and IDSW (Identity Switch)) of multiple objective tracking algorithms.

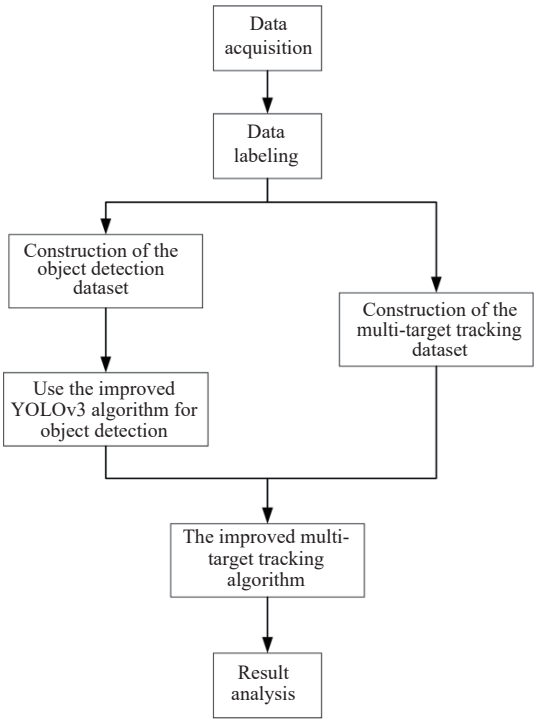


Figure 1 Technical route of this experiment

2.3 Dataset production

2.3.1 Object detection dataset

Two datasets were constructed to train the object detection

algorithm and verify the effect of multiple object tracking. The object detection dataset is shown in Figure 2. The image data used in the object detection dataset were collected every 10 s, and the collection periods included 10:30 a.m., 2:00 p.m., and 9:30 p.m. The LabelImg was used to label the yellow feather broilers, and the corresponding label information was stored in XML format. The label information includes the coordinates of the two ends of the rectangular box's diagonal, which reflects the size and position of a yellow feather broiler. A total of 500 images were used to construct the dataset. There were 45 yellow feather broilers in each image. The training dataset and the testing dataset were divided by 9:1, so the number of objects in the training dataset was 20,250, and the number of objects in the testing dataset was 2,250.

In order to enhance the generalization ability of the model, Mosaic data augmentation[24] was adopted after the object detection dataset was constructed. Each time, four images were selected. The selected images were processed by random flipping, random zooming, color gamut change, etc., and finally stitched together into one picture with a size of 416×416 pixels. Mosaic data augmentation enriched the object detection dataset. The operation of random scaling also added many small objects, which enhanced the robustness of the model. The sample of the object detection dataset after Mosaic data augmentation is shown in Figure 3.

2.3.2 Multiple object tracking dataset

This experiment used the surveillance video in the actual scenario to verify the tracking effect of the proposed method. The selected video clip had a high degree of activity and a significant change in the position of the yellow feather broiler flock as the test video. The length of the test video was 300 consecutive frames. The DarkLabel was used to label the test video. Different individuals were distinguished by different mark numbers in the labeling process, as shown in Figure 4.

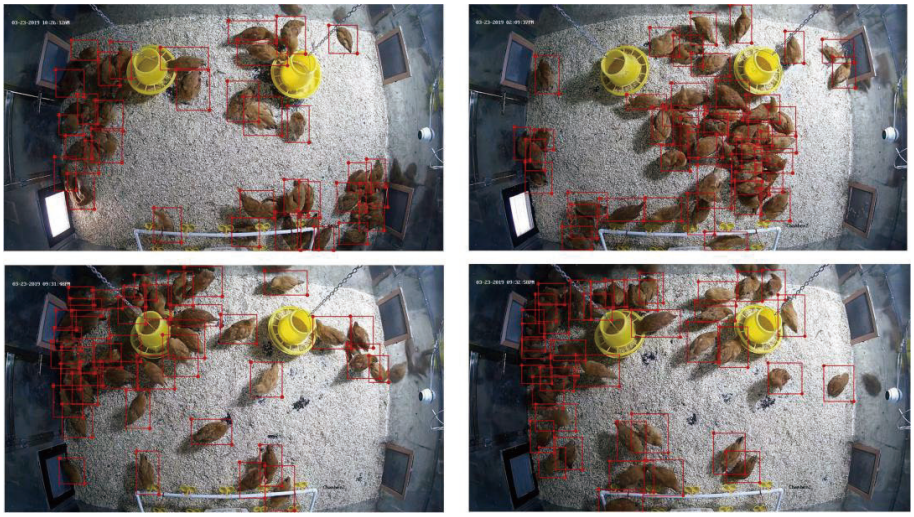


Figure 2 The object detection dataset

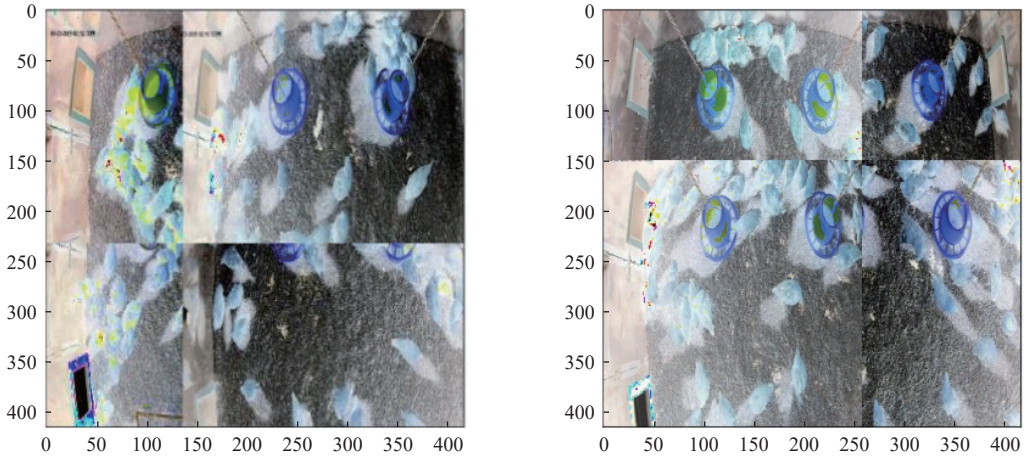


Figure 3 The sample images with 416×416 pixels after Mosaic data augmentation

Frame 1 Frame 300

Figure 4 Multiple object tracking dataset

2.4 Design of multiple object tracking of yellow feather broilers

This study used Deep SORT to achieve multiple object tracking of yellow feather broilers. Deep SORT is a multiple object tracking algorithm based on object detection. Compared with SORT, which is greatly affected by the influence of occlusion and reappearance[22], Deep SORT introduces a deep appearance descriptor. In the process of tracking, Deep SORT extracts the apparent features of objects for nearest neighbor matching, which improves the tracking effect in the presence of occlusion and reduces the problem of ID switching. As broilers in the broiler chamber are frequently occluded, Deep SORT has a better tracking effect than other algorithms. The flowchart of the multiple object tracking algorithm proposed in this paper is shown in Figure 5, which mainly includes four steps.

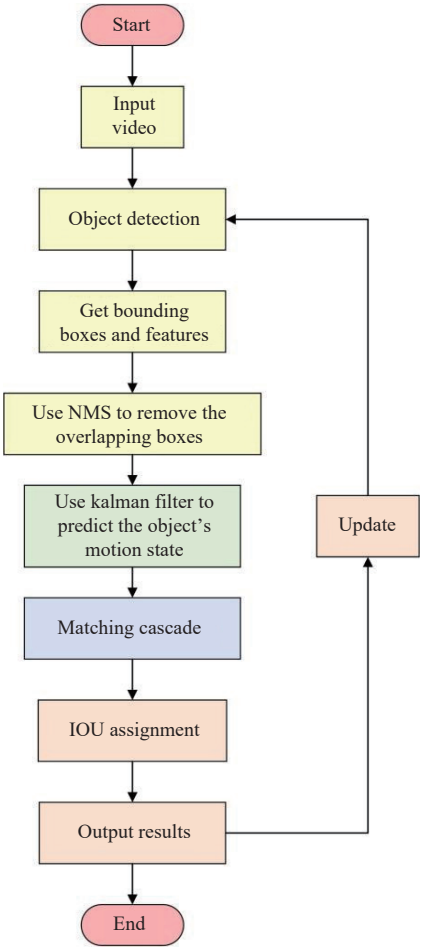


Figure 5 Flowchart of the multiple object tracking algorithm

(1) Use OpenCV to process the video into video frames, then use the improved YOLOv3 algorithm to obtain the object's position and depth feature. Use NMS (Non-Maximum Suppression) to remove the overlapping boxes and get the final detection result. The improved YOLOv3 algorithm will be introduced in detail in Section 2.5.

(2) Construct a motion model through a Kalman filter to predict the object's motion state. The center coordinates of the bounding box (μ, v), the aspect ratio γ, the height h, and their respective velocities in image coordinates $(\dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$ were used as eight parameters to describe the position and motion information of the object.

(3) Based on the obtained appearance information and movement information, Matching Cascade was used for measurement-to-track association.

Motion information association: Mahalanobis distance was used to calculate the distance between predicted Kalman states and newly arrived measurements, as shown in Equation (1).

$$d^{(1)}(i,\ j) = \left(d_j - y_i\right)^T S_i^{-1} \left(d_j - y_i\right) \tag{1}$$

where, the projection of the i^{th} track distribution is denoted into measurement space by $(y_i,\ S_i)$, and the j^{th} bounding box detection by d_j. If the obtained Mahalanobis distance is less than the specified threshold $t^{(1)}$, then the motion state association could be regarded as successful.

Appearance information association: For each bounding box detection d_j an appearance descriptor r_j with $\|r_j\| = 1$ was computed. A gallery $R_k = \left\{r_k^{(i)}\right\}_{k=1}^{L_k}$ of the last L_k=100 associated appearance descriptors for each track k was kept. When the smallest cosine distance between the i^{th} track and j^{th} detection in appearance space is less than the specific threshold $t^{(2)}$, then the appearance information could be considered to be successfully associated. The calculation of the smallest cosine distance is shown in Equation (2).

$$d^{(2)}(i,\ j) = \min\{1 - r_j^T r_k^{(i)} |\ r_k^{(i)} \in R^i\} \tag{2}$$

The result of linear weighting of the two metrics is used as the final measurement $c_{i,j}$. Only when $c_{i,j}$ was within the gating region of both metrics, the association could be called admissible, as shown in Equation (3).

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j) \tag{3}$$

(4) After Matching Cascade, perform the IoU (Intersection over Union) assignment for unmatched detection, unconfirmed tracks, and unmatched tracks. If the matching is successful, the Kalman filter will be updated. Otherwise, a new track will be established for the unmatched detections, while a maximum age will be set for the unmatched tracks. After three consecutive hits within the maximum age, the tracks will be changed from an unconfirmed state to a confirmed one.

2.5 Object detection

The object detection algorithm is a very important part of Deep SORT, which is related to tracking accuracy and speed. The YOLOv3 was chosen as the fundamental network and proposed an object detection algorithm that was more suitable for detecting yellow feather broilers in flat breeding chambers.

2.5.1 YOLOv3

As a mature one-stage object detection algorithm, YOLOv3 has a simple structure and good detection accuracy compared with the two-stage detection algorithm. In the detection process, YOLOv3 divides the input image into S×S grids. If the center coordinates of the object to be measured fall on a certain grid, the grid is responsible for detecting the object. Each grid predicts B bounding boxes. Each bounding box corresponds to (4+1+C) predicted values, where '4' represents the width, height, and center coordinates of the bounding box, '1' represents the confidence that the bounding box has an object, and 'C' represents the probability that the object belongs to each of 'C' categories. The YOLOv3 detection algorithm ultimately yields a tensor of size S×S×[B×(4+1+C)].

2.5.2 Improved YOLOv3 model

(1) MobileNetV2

The MobileNetV2[25] is used to replace Darknet53 as the backbone. MobileNetV1 uses DW (DepthWise convolutions) to trade-off between computation and accuracy effectively.

MobileNetV2 is the upgraded version of MobileNetV1, with two improvements:

a) Introduce inverted residual structure. The inverted residual block takes a low-dimensional compressed representation as input and expands it to a high dimension. It enhances the propagation of the gradient and dramatically reduces the memory required in the inference process.

b) Introduce the idea of a linear bottleneck where the last convolution of a residual block has a linear output before it is added to the initial activations.

The comparison of module structure between MobileNetV1 and MobileNetV2 is shown in Figure 6.

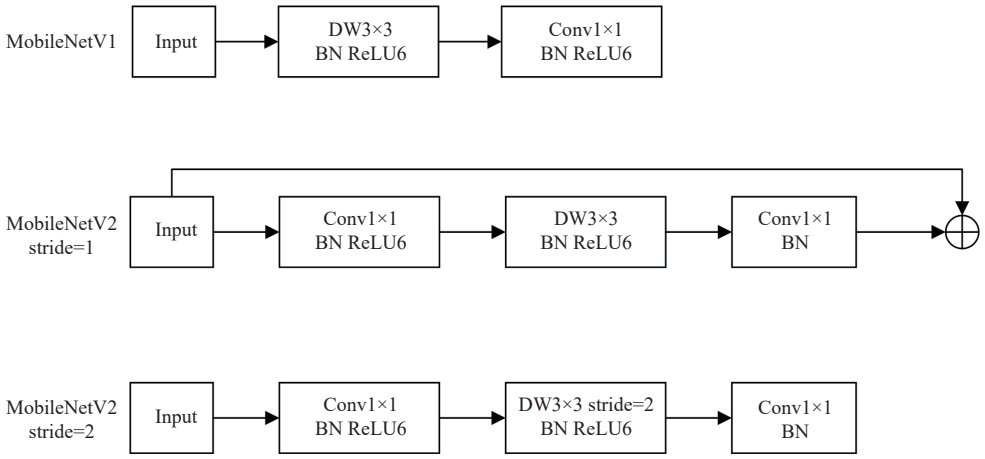


Figure 6 Comparison of module structure between MobileNetV1 and MobileNetV2

(2) DRSN

DRSN[26] is a deep learning method for noisy data. It solves the problem that the effect of the deep learning algorithm will be reduced when the data contains noise or redundant information. DRSN integrates the deep residual network, soft thresholding, and attention mechanism. Soft thresholding removes the features whose absolute value is less than a certain threshold and shrinks the absolute value features from this threshold toward zero. Equation (4) is shown as follows.

$$y = \begin{cases} x-\tau, & x > \tau \\ 0, & -\tau \le x \le \tau \\ x+\tau, & x < -\tau \end{cases} \tag{4}$$

where, x represents the input feature, y represents the output feature, and τ represents the threshold. The threshold is adaptively learned by the attention mechanism. The important features extracted by the attention mechanism are retained through soft thresholding, which strengthens the ability of the network to extract useful features from noisy signals. The building unit of DRSN and the inverted residual block of MobileNetV2 are combined to form a new block entitled DRSN-Inverted residual. The structure of the DRSN-Inverted residual is shown in Figure 7. As a result, a lightweight backbone entitled DRSN-MobileNetV2 with a stronger feature extraction capability was constructed.

(3) Feature fusion and attention mechanism

YOLOv3 uses K-means Clustering to obtain nine anchors. Every three anchors are treated as a group, as the default anchors for feature maps of three scales. Considering the size distribution of yellow feather broilers is concentrated, the sizes of anchors obtained by K-means Clustering will be similar. Therefore, the receptive field may not fit the object since objects of similar size will be forced to be sent to different layers for prediction. Thus, the CEM (Context Enhancement Module) of ThunderNet[14] was used to create a feature fusion network with a single output that can aggregate information of various scales. While fusing the feature maps of the three output layers of YOLOv3, the receptive field of the model was expanded as well. Feature maps of different scales contain different semantic information, but the feature fusion method used by CEM is to add the feature maps directly, ignoring the differences in each output layer. Therefore, the attention mechanism is introduced, including CBAM (Convolutional Block Attention Module)[27] and SE (Squeeze-and-Excitation)[28] block. The structures of CBAM and SE blocks are shown in Figure 8.

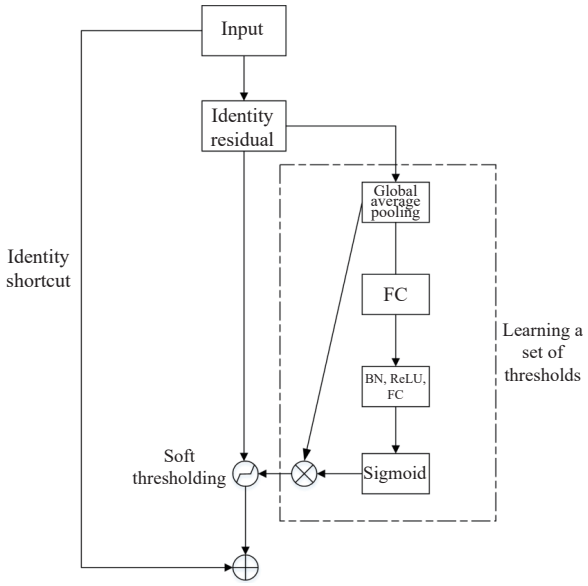


Figure 7 Structure of DRSN-Inverted residual

CBAM was inserted into the three output layers: a channel attention module and a spatial attention module were inserted in series into the original network. CBAM blended cross-channel information and spatial information to extract features. Meanwhile, an SE block was used to fuse the information extracted from the high level of the feature pyramid to the shallow layer in a multiplying manner to realize the semantic extraction of the shallow layer guided by the semantic information of the deep layer.

An improved YOLOv3 algorithm, which combined the above three improvement measures, was proposed. The comparison between the proposed model structure and the YOLOv3 network structure is shown in Figure 9.

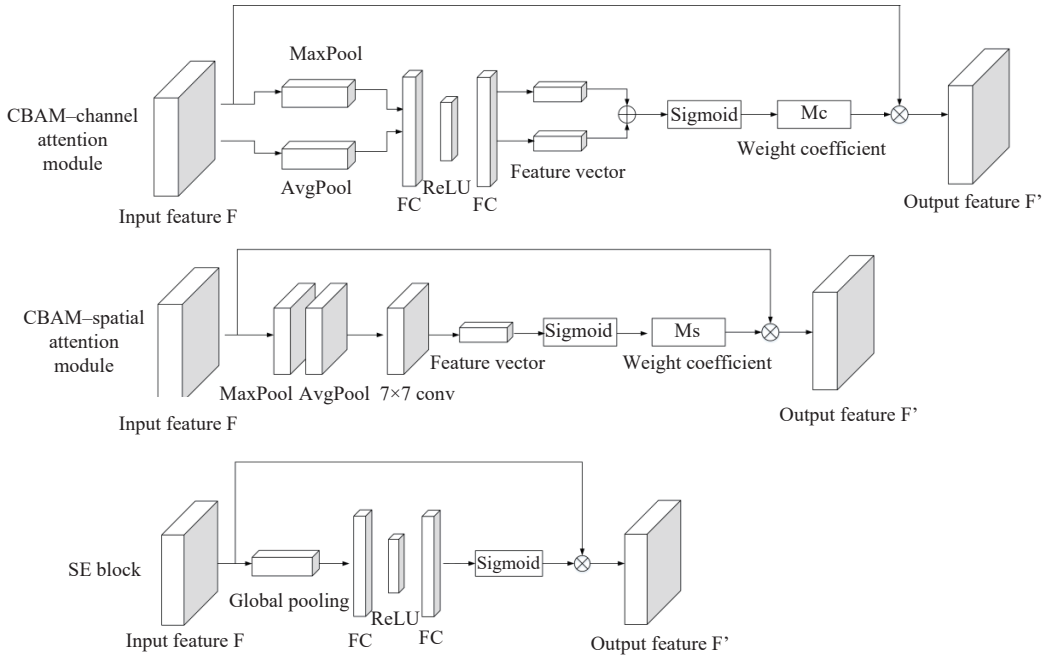


Figure 8 Structure of CBAM and SE block

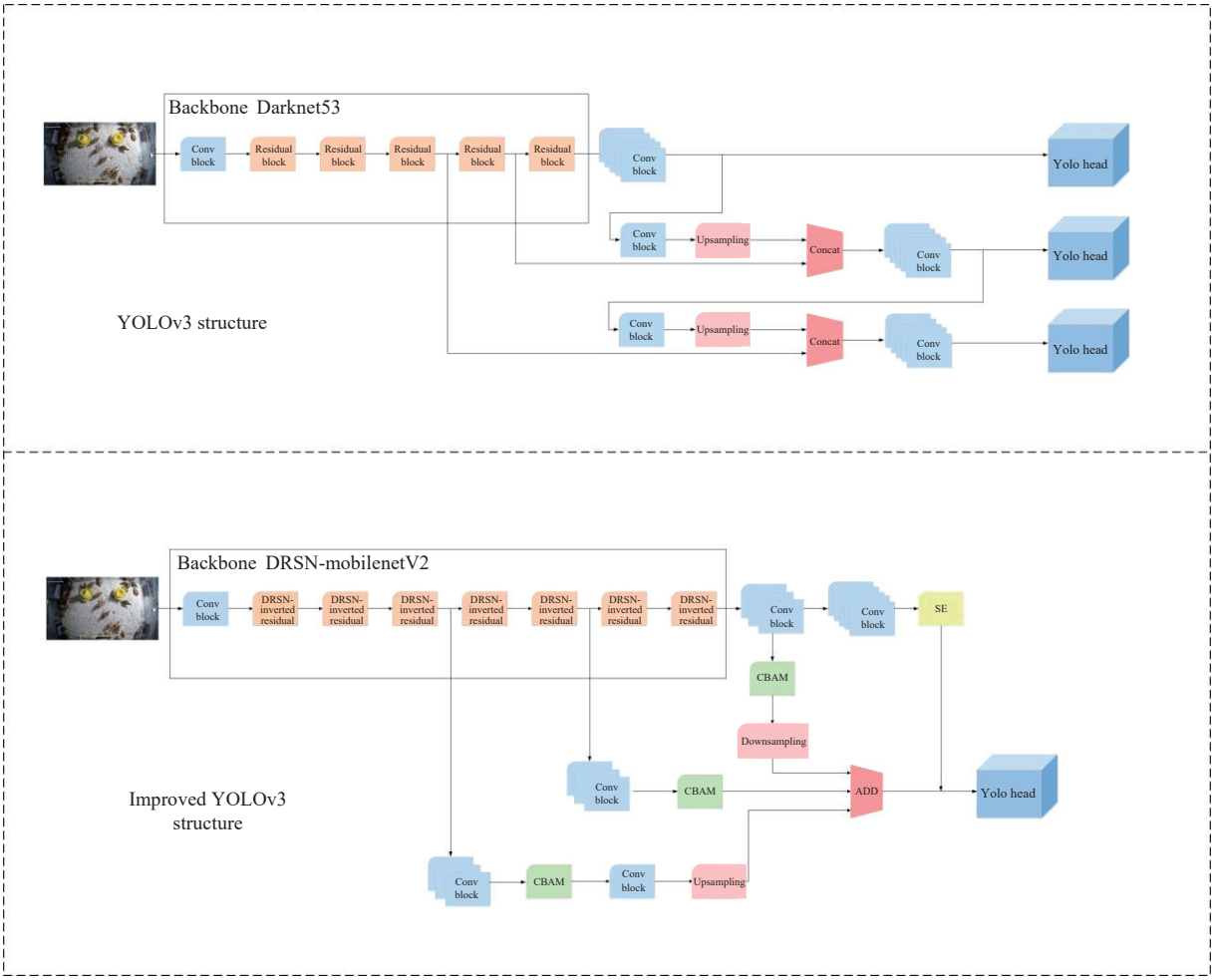


Figure 9 Comparison between the proposed model structure and the YOLOv3 network structure

3 Results and discussion

3.1 Experimental conditions and configuration

The software platforms used in this experiment are Pycharm2019.3, OpenCV-Python=4.5.1, torch1.7.0+cuda11.0, and torch vision 0.8.1+cuda11.0.

The hardware platform is Dell Inspiron15-7572 (Intel Core i5 8th Gen, Computer Memory 8 GB). Connect to the cloud host during training; the cloud host is GeForce RTX 2080 Ti with 11GB Video Memory.

3.2 Detection results

3.2.1 Network initialization parameters

The initialization parameters of the improved YOLOv3 network are listed in Table 1. The network input size was set to 416×416 pixels. The batch size was set to 16, considering the memory constraints of the server. Two hundred training steps were used to train the proposed algorithm, and the parameters of the backbone network were frozen in the first one hundred training steps. To avoid the local minima in gradient descent, Cosine Annealing[29] was used to adjust the learning rate during training. Cosine Annealing is a type of learning rate schedule that has the effect of starting with a large learning rate that is rapidly decreased to a minimum value before being increased rapidly again.

Table 1 The initialization parameters of the improved YOLOv3 network

Size of input images	Batch	Initiallearning rate	Learning rate schedule	Trainingsteps
416 × 416	16	0.001	Cosine Annealing	200

3.2.2 Evaluation metrics

This experiment selected seven indicators to quantitatively evaluate the performance of the algorithm, of which there are four accuracy indicators: P (Precision), R (Recall), F1 Score[30], and mAP[31], three model complexity indicators: FPS, GFLOPs (Giga Floating-point Operations) and Params.

(1) P and R

In object detection, P and R are the basic evaluation indicators. P is defined as the proportion of correctly detected objects in all detected objects, whereas R refers to the proportion of correctly detected objects in all positive samples. The equations of these two indicators are as follows.

$$P = \frac{TP}{TP + FP} \times 100\% \quad (5)$$

$$R = \frac{TP}{TP + FN} \times 100\% \quad (6)$$

where, TP (True Positive) is the number of yellow feather broilers correctly detected, FP (False Positive) is the number of the non-yellow feather broilers that are identified as yellow feather broilers, and FN (False Negative) is the number of yellow feather broilers that are identified as the non-yellow feather broilers.

(2) mAP and F1 Score

mAP represents the average value of the AP (Average Precision) of each category. The mAP evaluates the quality of the model in all categories and is an indicator that considers P and R. Here, since the detection objects belong to only one category, mAP = AP, where AP is the area under the P-R curve, as shown in Equation (7).

$$AP = \int_0^1 P(R)dR \quad (7)$$

The F1 Score is also an indicator to evaluate the performance of the proposed model. The calculation equation is shown in Equation (8). The higher the F1 Score is, the better the model performance will be.

$$F1 = \frac{2P \times R}{P + R} \quad (8)$$

(3) FPS

In addition to the accuracy of detection, FPS is another commonly used indicator to evaluate the speed of object detection, which represents the number of images that can be processed per second.

(4) GFLOPs and Params

1 GFLOPs is equal to 1 billion floating-point operations, which can be understood as the amount of calculation required by the model. Params refer to the total number of parameters that the model needs to train. GFLOPs and Params are used to measure the complexity of the model.

3.2.3 Comparison of different algorithms

The proposed model was compared with Faster R-CNN[18], SSD[10], and YOLOv3[9] to verify the performance of the model. Faster R-CNN is a two-stage detection algorithm with higher accuracy than one-stage detection algorithms, but it cannot meet the real-time requirements. Faster R-CNN is an improvement of Fast R-CNN, which integrates the acquisition of region proposals in the CNN, realizes end-to-end training and testing, and dramatically improves its efficiency. YOLOv3 and SSD are one-stage detection algorithms. SSD draws on the idea that YOLO transforms the object detection task from a classification problem into a regression problem, eliminating the region proposal process, which significantly reduces the inference time. At the same time, SSD adds the feature pyramid to the network and makes predictions on feature maps of different scales, which improves the performance of detection. YOLOv3 is an improvement of YOLO and YOLO9000. Compared with most one-stage detection algorithms, it has higher detection accuracy. YOLOv3 uses Darknet53 as the backbone to extract image features. YOLOv3 uses upsampling to fuse multi-scale feature maps so that the feature maps have rich semantic information. Meanwhile, the idea of multi-scale prediction is introduced, which realizes prediction on feature maps of three different scales. The comparison between the proposed model and the three object detection models listed above could prove the superiority of the proposed model in detection accuracy and inference speed.

Figure 10 shows the curves of the total loss of YOLOv3 and the proposed model with the number of iterations during training. The initialization parameters of the two models are the same, as given in Section 3.2.1.

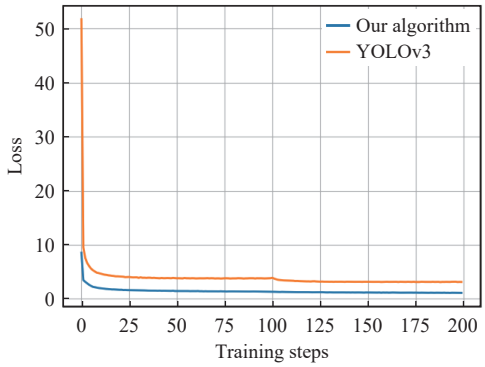


Figure 10 Curves of the total loss of YOLOv3 and the proposed model with the number of iterations during training

It can be seen from Figure 10 that the model has a faster convergence speed and lower convergence loss than YOLOv3. The total loss of both models gradually decreases with the iteration of the models and eventually stabilizes. The convergence loss of the YOLOv3 model is stable at about 3.1, and the convergence loss of the proposed model is stable at about 1.2, which is lower than that of YOLOv3, indicating that the improvement measures have improved the performance of the model.

The P-R curve of each model during testing is shown in

Figure 11. In addition, Table 2 shows the various indicators of the experimental results of each model, including P, R, F1 Score, mAP, and FPS.

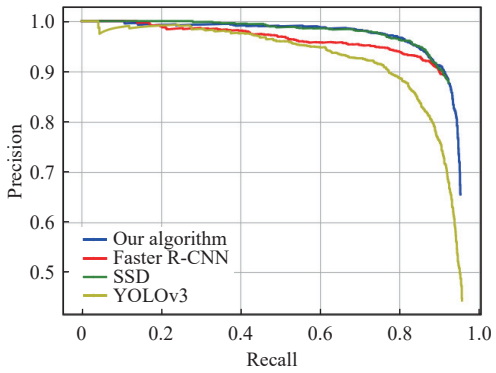


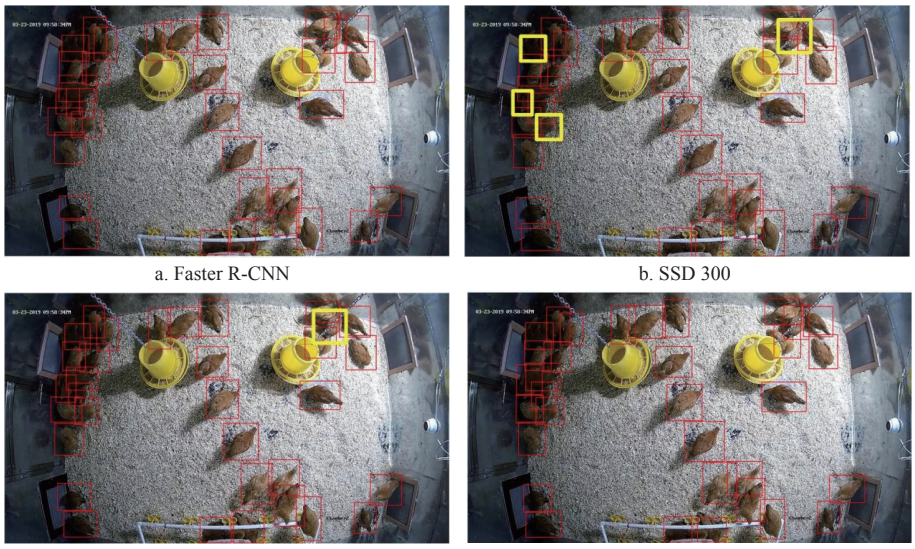
Figure 11 P-R curve of each model during testing

The proposed model has the highest F1 Score and mAP based on the above detection results, and it is the only model whose P and R both reach more than 90%, indicating that the proposed model has a better performance than the other three models. The P-R curve in Figure 11 also confirms this conclusion. The F1 Score of the proposed model is 0.07 higher than that of YOLOv3, reaching 0.91. Compared with YOLOv3, P and R are 9.95% and 3.06% higher, respectively, reflecting the overall performance improvement of the model compared to the original network. In terms of inference speed, the proposed model is the only model with an FPS exceeding 25 fps (frames per second), which means better real-time detection performance on the GPU (Graphics Processing Unit).

Table 2 Comparison of the detection results of different models

	F1 Score	P/%	R/%	mAP/%	FPS/fps
Faster R-CNN	0.90	88.57	**91.84**	88.98	0.59
SSD 300	0.89	**95.84**	82.81	90.89	11.32
YOLOv3	0.84	80.54	87.76	89.32	5.89
The proposed model	**0.91**	90.49	90.82	**93.21**	**29.48**

The detection results of each model on the testing dataset are shown in Figure 12. The yellow boxes outline the missed detections in the test results. Both the SSD model and the YOLOv3 model have missed detections. Most of the missed broilers are those with a more severe cover or smaller size. The phenomenon of missed detection of the SSD model is more severe than YOLOv3. The Faster R-CNN model and the proposed model have detected all broilers without missing any detections.



a. Faster R-CNN b. SSD 300

c. YOLOv3 d. Our algorithm

Figure 12 The detection results of each model on the testing dataset

3.2.4 Influence of the backbone

MobileNetV2 was used instead of Darknet53 as the backbone. In the case of keeping the structure of the rest of the model consistent, the complexity of the models with the backbone of MobileNetV2 and Darknet53 was evaluated. The evaluation results of model complexity are listed in Table 3.

Table 3 Evaluation results of the complexity of the models with the backbone of MobileNetV2 and Darknet53

Backbone	GFLOPs	Params	FPS/fps
Darknet53	25.87	43.01M	9.31
MobileNetV2	**1.68**	**4.34M**	**29.48**

The accuracy of the two models was also evaluated. The evaluation results are listed in Table 4.

It can be seen from Table 3 that the FPS of the model whose backbone is Darknet53 is only 9.31 fps, and the FPS rises to 29.48 fps after replacing it with MobileNetV2. At the same time, the GFLOPs and Params of the model after the replacement are 6.49% and 10.09% of the original model. It can be found from the further analysis of the evaluation results in Table 3 and Table 4 that although Darknet53 has a more complex structure, the accuracy of the model after replacing the backbone has not significantly decreased, and even the R and the mAP have increased slightly.

Table 4 Evaluation results of the accuracy of the models with the backbone of MobileNetV2 and Darknet53

Backbone	F1 Score	P/%	R/%	mAP/%
Darknet53	0.90	**92.96**	89.38	92.23
MobileNetV2	0.90	89.90	**89.94**	**92.56**

3.2.5 Influence of the DRSN

DRSN was introduced to enhance the backbone's ability to extract useful features. In order to study the impact of the DRSN on

the model, the output features of the original model and the modified model of the same image (Figures 13a and 13d) were compared. The t-SNE algorithm[32] was used to reduce the dimension of the output features, and then they were mapped to a two-dimensional space for visualization. The results are shown in Figures 13b, 13c, 13e and 13f.

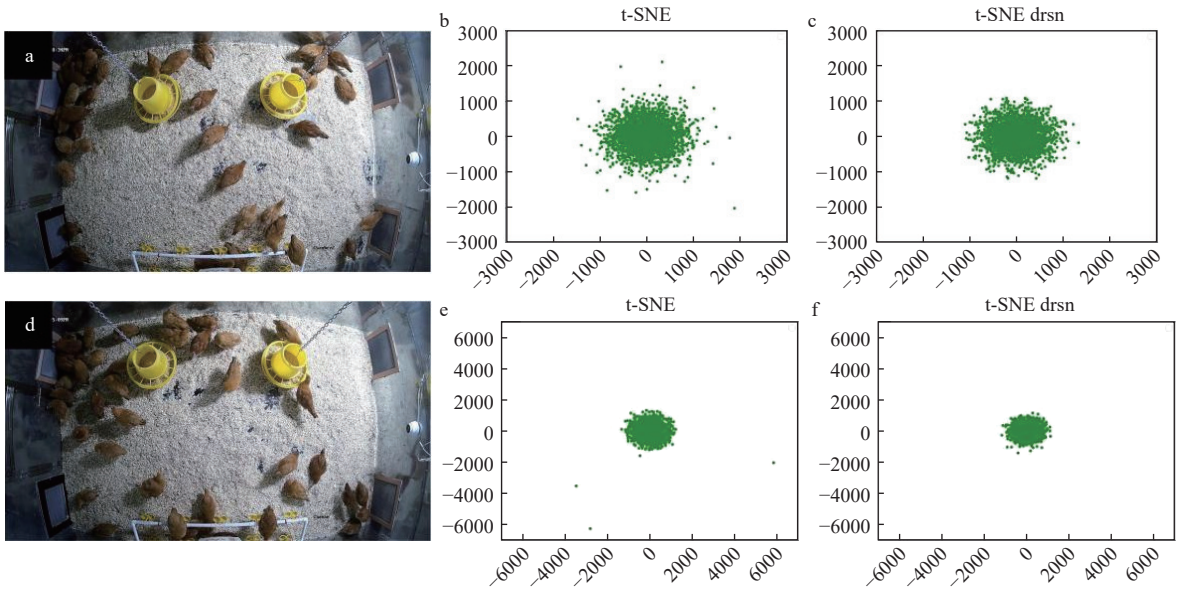


Figure 13 Visualization results of output features after dimensionality reduction: (a)(d) input images; (b)(e) feature visualization of the model before the introduction of DRSN; (c)(f) feature visualization of the model after the introduction of DRSN

Comparing the visualized features in Figure 13, the distribution of the output features after DRSN processing is closer so that it can be better distinguished from the background information. In addition, Figures 13b and 13e show that there is noise information in the features before DRSN is introduced. DRSN eliminates part of the noise information, reduces interference, and improves the accuracy of the final detection results.

The models before and after the introduction of DRSN were tested on the testing dataset to verify the improvement of the model performance. Part of the test results are shown in Figure 14.

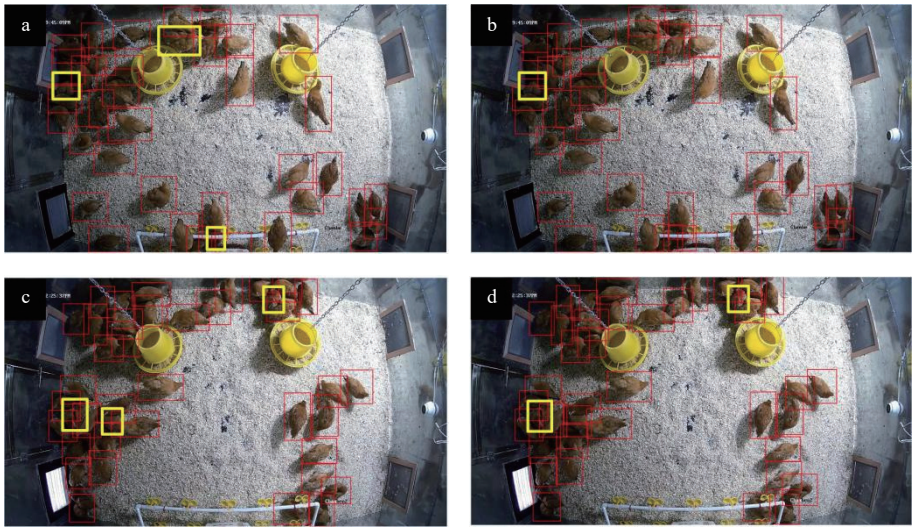


Figure 14 Test results of the models: (a)(c) the model before the introduction of DRSN; (b)(d) the model after the introduction of DRSN

The yellow boxes outline the missed detections in the test results. According to the comparison of the test results after the introduction of DRSN, due to the tighter distribution of features, the features of the yellow feather broilers can be better distinguished from other information. Therefore, the model has a better recognition effect on some yellow feather broilers that are more severely occluded than before. As a result, the number of missed detections is reduced, and the detection accuracy of the model is improved.

The accuracy of the detection results of the two models was further evaluated, and the results are listed in Table 5.

Table 5 Evaluation results of the accuracy of the models before and after the introduction of DRSN

	F1 Score	P/%	R/%	mAP/%	FPS/(f·s^{-1})
Without DRSN	0.90	89.90	89.94	92.56	**31.34**
With DRSN	**0.91**	**90.49**	**90.82**	**93.21**	29.48

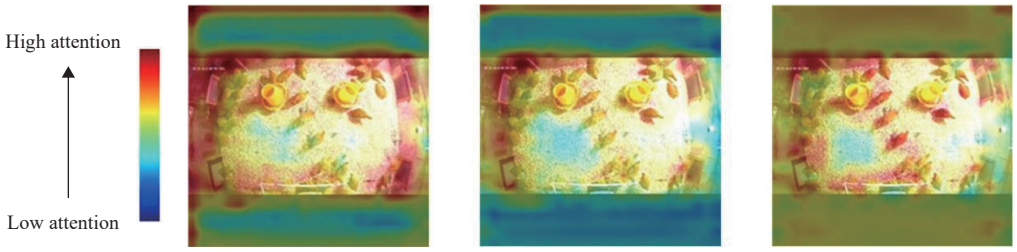
It can be seen from Table 5 that after the introduction of DRSN, various indicators of the model have been improved, while the detection speed is only reduced by 5.93%, which does not affect the real-time performance of detection.

3.2.6 Influence of attention mechanism

The feature maps output by different models were visualized as CAM (Class Activation Maps)[33] to verify the improvement of the model performance by attention mechanism. CAM can clearly show the region of the image that the model focuses on when making a prediction.

The feature visualization results of the original model, the model added with CBAM, and the model added with CBAM and the SE block were compared, as shown in Figure 15.

Through the comparison, a considerable part of the features extracted by the original model is covered in the background, so the feature extraction is not performed effectively. After adding CBAM, the model's attention to the background is reduced, but it does not focus on the yellow feather broilers. Since the semantic extraction of the shallow layer is guided by the semantic information of the deep layer of the network after the SE block is added, the features extracted by the model cover more of the objects that need to be detected.



a. No attention mechanism b. CBAM c. CBAM+SE

Figure 15 Feature visualization results of the original model, the model added with CBAM, and the model added with CBAM and the SE block

The detection results of the original model, the model with CBAM, and the model with CBAM and the SE block were evaluated, and the results are listed in Table 6.

Table 6 Accuracy comparison of models corresponding to different attention mechanisms

	F1 Score	P/%	R/%	mAP/%
No attention mechanism	0.89	89.55	89.28	92.19
CBAM	0.90	89.90	89.94	92.70
CBAM+SE	**0.91**	**90.49**	**90.82**	**93.21**

Table 6 shows that with the addition of CBAM and the SE block, the detection accuracy of the models is gradually improved. Compared with the model with no attention mechanism, the mAP of the final model increases by 1.02%, and the F1 Score increases by 0.2. Moreover, after the introduction of the attention mechanism, the features extracted by the model cover more broilers, which improves the accuracy of detection.

3.3 Tracking results

3.3.1 Evaluation metrics

This experiment selected five indicators to evaluate the effect of multiple object tracking[34]:

(1) IDSW represents the number of identity switches. The lower the IDSW indicates, the better the model performance will be.

(2) MOTA combines three error sources: false positives, missed targets, and identity switches. The higher the MOTA means, the better the model performance will be. The calculation equation is shown as Equation (9).

$$\mathrm{MOTA} = 1 - \frac{\sum\left(\mathrm{A_{FP}} + \mathrm{A_{FN}} + \mathrm{A_{ID}}\right)}{\sum \mathrm{A_{GT}}} \tag{9}$$

where, $\mathrm{A_{FP}}$ is the number of false positives, $\mathrm{A_{FN}}$ is the number of false negatives, $\mathrm{A_{ID}}$ is the number of ID Switch, and $\mathrm{A_{GT}}$ is the number of targets.

(3) IDF1(Identification F1 Score) represents the ratio of correctly identified detections over the average number of ground-truth and computed detections. It is the first default indicator used to evaluate the quality of the tracker. The calculation equation is shown as Equation (10).

$$\mathrm{IDF1} = \frac{2\mathrm{IDTP}}{2\mathrm{IDTP} + \mathrm{IDFP} + \mathrm{IDFN}} \tag{10}$$

where, IDTP is the positive sample that is correctly identified, IDFP is the negative sample that is incorrectly identified, and IDFN is the positive sample that is incorrectly identified.

(4) P and R. P represents the percentage of correctly matched detections to total detections. R represents the percentage of correctly matched detections to ground-truth detections。

3.3.2 Evaluation of tracking results

The proposed model was compared with the YOLOv3-Deep SORT model in terms of five selected evaluation metrics for tracking results. The evaluation results are shown in Table 7.

Table 7 Comparison of tracking results of different models

	IDSW	MOTA/%	IDF1/%	P/%	R/%
YOLOv3-Deep SORT	37	51.1	68.0	79.0	70.1
The proposed model	**14**	**54.0**	**72.7**	**79.9**	**72.3**

The MOTA of the proposed model is 54%, and IDF1 is 72.7%, which are respectively 2.9% and 4.7% higher than the YOLOv3-Deep SORT model. IDSW is reduced by 23, which is 37.8% of the YOLOv3-Deep SORT model. P and R have been improved as well. Experimental data proves that the model is superior to the YOLOv3-Deep SORT model in detection and tracking.

The tracking effect of the model on yellow feather broilers in different states was further analyzed. Broilers with representative behaviors in the test video were selected, including yellow feather broilers in a flapping state, running state, gathering state, eating state, and drinking state, to study the tracking results of the model on them at different times. The tracking results are shown in Figure 16.

Figure 16 shows that no matter what state the yellow feather broiler is in, there is no target loss during the tracking, and the ID number of the target has not changed, indicating that the proposed model can track the yellow feather broilers in different states stably and maintain good robustness in the complex flat breeding environment. It can provide technical support to perceive the behavior of yellow feather broilers and study the relationship

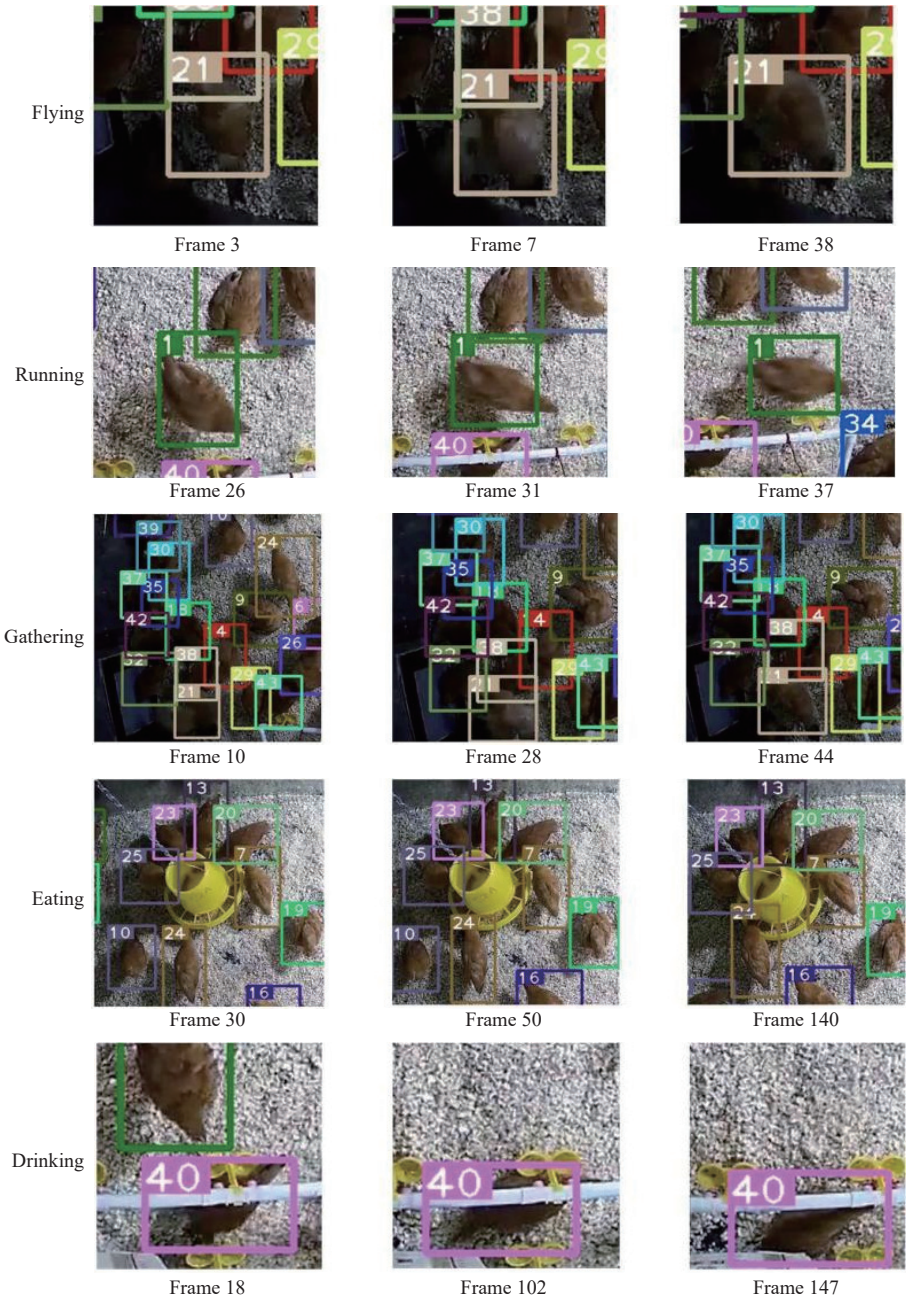


Figure 16 Tracking effect of the proposed model on yellow feather broilers in different states

4 Conclusions

(1) In this paper, an improved YOLOv3 algorithm was proposed. MobileNetV2 was used to replace the backbone of YOLOv3 to improve the inference speed of the detection module. The DRSN was integrated with the feature extraction module of MobileNetV2 to enhance the feature extraction capability of the backbone. The feature fusion network was redesigned, combined with the attention mechanism, to realize the adaptive learning of multi-scale features of the objects. Experimental results show that the improved YOLOv3 algorithm has an mAP of up to 93.2%, which exceeds other object detection algorithms, and has an FPS reached 29 fps, which is almost five times that of YOLOv3.

(2) The improved object detection algorithm was combined with Deep SORT to realize the multiple object tracking of yellow feather broilers. Experimental data prove that the proposed algorithm is superior to the YOLOv3-Deep SORT algorithm regarding detection and tracking: MOTA and IDF1 are increased by 2.9% and 4.7%, respectively. The IDSW of the proposed algorithm is 37.8% of the YOLOv3-Deep SORT.

(3) The algorithm can achieve stable tracking of yellow feather broilers in different states by analyzing the tracking results of the test video. The behavior of a yellow feather broiler can reflect its health status. In the future, the tracking results can be further analyzed to establish the quantitative relationship between health status and behavior statistics to find the abnormalities of yellow feather broilers in time. The proposed algorithm realizes the multiple object tracking of yellow feather broilers, which can be used as the basis of behavior perception and provide technical support for the behavior analysis of yellow feather broilers, which is of great significance to the breeding of yellow feather broilers.

(4) This algorithm will be recommended to be used in the system of flat breeding chambers to realize real-time and dynamic detection and tracking of yellow feather broilers. Since the GFLOPs and Params of the improved model are 6.49% and 10.09% of the original model, respectively, its calculation speed is enough to run

on the real system.

Acknowledgement

This research was funded by Jiangsu Agriculture Science and Technology Innovation Fund (Grant No. CX(21)3058), Xuzhou Key Research and Development Project (Modern Agriculture) (Grant No. KC21135) and International Science and Technology Cooperation Program of Jiangsu Province (Grant No. BZ2023013).

[References]

[1] Mottet A, Tempio G. Global poultry production: current state and future outlook and challenges. World's Poultry Science Journal, 2017; 73(2): 245–256.

[2] Xiao L, Ding K, Gao Y, Rao X. Behavior-induced health condition monitoring of caged chickens using binocular vision. Computers and Electronics in Agriculture, 2019; 156: 254–262.

[3] Fujii T, Yokoi H, Tada T, Suzuki K, Tsukamoto K. Poultry tracking system with camera using particle filters. 2008 IEEE International Conference on Robotics and Biomimetics, Bangkok, Thailand, Feb. 22-25, 2009; pp.1888–1893.

[4] Ahrendt P, Gregersen T, Karstoft H. Development of a real-time computer vision system for tracking loose-housed pigs. Computers and Electronics in Agriculture, 2011; 76(2): 169–174.

[5] Nakarmi A D, Tang L, Xin H. Automated tracking and behavior quantification of laying hens using 3D computer vision and radio frequency identification technologies. Transactions of the ASABE, 2014; 57(5): 1455–1472.

[6] Mittek M, Psota E, Carlson J D, Pérez L C, Schmidt T, Mote B. Tracking of group-housed pigs using multi-ellipsoid expectation maximisation. IET Computer Vision, 2018; 12(2): 121–128.

[7] Redmon J, Divvala S, Girshick R B, Farhadi A. You Only Look Once: unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, Jun. 27-30, 2016; pp.779–788.

[8] Redmon J, Farhadi A. YOLO9000: better, faster, stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, Jul. 21-26, 2017; pp.6517–6525.

[9] Redmon J, Farhadi A. YOLOv3: an incremental improvement. arXiv, 2018;doi: 10.48550/arXiv.1804.02767

[10] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, et al. SSD: Single Shot MultiBox Detector. European Conference on Computer Vision (ECCV), Amsterdam, Netherlands, Oct. 11-14, 2016; pp.21–37.

[11] Fu C-Y, Liu W, Ranga A, Tyagi A, Berg A C. DSSD: Deconvolutional Single Shot Detector. arXiv, 2017;doi: 10.48550/arXiv.1701.06659.

[12] Li Z, Zhou F. FSSD: Feature fusion single shot multibox detector. arXiv, 2017;doi: 10.48550/arXiv.1712.00960.

[13] Lin T-Y, Goyal P, Girshick R B, He K, Dollár P. Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020; 42(2): 318–327.

[14] Qin Z, Li Z, Zhang Z, Bao Y, Yu G, Peng Y, et al. ThunderNet: towards real-time generic object detection on mobile devices. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), Oct. 27 - Nov. 2, 2019; pp.6717–6726.

[15] Sun Q, Wu T, Zou X, Qiu X, Yao H, Zhang S, et al. Multiple object tracking for yellow feather broilers based on foreground detection and deep learning. INMATEH-Agricultural Engineering, 2019; 58(2): 155–166.

[16] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, Jun. 23-28, 2014; pp.580–587.

[17] Girshick R. Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, Dec. 7-13, 2015; pp.1440–1448.

[18] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015; 39(6): 1137–1149.

[19] Sun L Q, Zou Y B, Li Y, Cai Z D, Li Y, Luo B, et al. Multi target pigs tracking loss correction algorithm based on Faster R-CNN. Int J Agric & Biol Eng, 2018; 11(5): 192–197.

[20] Lin C-Y, Hsieh K-W, Tsai Y-C, Kuo Y-F. Monitoring chicken heat stress using deep convolutional neural networks. 2018 ASABE Annual International Meeting, Detroit, Michigan, USA, Jul. 29 - Aug. 1, 2018;doi: 10.13031/aim.201800314.

[21] Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. Simple online and realtime tracking. 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, Arizona, USA, Sep. 25-28, 2016; pp.3464–3468.

[22] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, Sep. 17-20, 2017; 3645–3649.

[23] Yao H, Sun Q, Zou X, Wang S, Zhang S, Zhang S, et al. Research of yellow-feather chicken breeding model based on small chicken chamber. INMATEH-Agricultural Engineering, 2018; 56(3): 91–100.

[24] Bochkovskiy A, Wang C-Y, Liao H. YOLOv4: optimal speed and accuracy of object detection. arXiv, 2020;doi: 10.48550/arXiv.2004.10934.

[25] Sandler M, Howard A G, Zhu M, Zhmoginov A, Chen L-C. MobileNetV2: inverted residuals and linear bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, Jun. 18-23, 2018; pp.4510–4520.

[26] Zhao M, Zhong S, Fu X-y, Tang B, Pecht M. Deep residual shrinkage networks for fault diagnosis. IEEE Transactions on Industrial Informatics, 2020; 16: 4681–4690.

[27] Woo S, Park J, Lee J-Y, Kweon I S. CBAM: Convolutional Block Attention Module. European Conference on Computer Vision (ECCV), Munich, Germany, Sep. 10-13, 2018; 3–19.

[28] Hu J, Shen L, Albanie S, Sun G, Wu E. Squeeze-and-excitation networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020; 42(8): 2011–2023.

[29] Loshchilov I, Hutter F. SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv, 2017;doi: 10.48550/arXiv.1608.03983.

[30] Powers D. M. W. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv, 2020;doi: 10.48550/arXiv.2010.16061.

[31] Everingham M, Winn J. The Pascal visual object classes challenge 2012 (voc2012) development kit. Pattern Analysis, Statistical Modelling and Computational Learning, Tech., 2011.

[32] Maaten L, Hinton G E. Visualizing data using t-SNE. Journal of Machine Learning Research, 2008; 9(86): 2579–2605.

[33] Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, Jun. 27-30, 2016; pp.2921–2929.

[34] Milan A, Leal-Taixé L, Reid I, Roth S, Schindler K. MOT16: a benchmark for multi-object tracking. arXiv, 2016;doi: 10.48550/arXiv.1603.00831.