MULTIPLE OBJECT TRACKING FOR YELLOW FEATHER BROILERS BASED ON FOREGROUND DETECTION AND DEEP LEARNING

/

基于前景检测和深度学习的黄羽鸡多目标跟踪

Qiyue Sun [1)], Tinghui Wu [1)], Xiuguo Zou *[1) 2)], Xinfa Qiu [2)], Heyang Yao [1)], Shikai Zhang[1)], Yuning Wei [1)] [1]

[1)] College of Engineering, Nanjing Agricultural University / China;

[2)]School of applied meteorology, Nanjing University of Information Science and Technology / China

Tel: +862558606585; E-mail: xiuguozou@gmail.com

DOI: 10.35633/INMATEH-58-17

***Keywords:** broiler, multiple object tracking, foreground detection, YOLOv3*

ABSTRACT

In view of the two problems existing in the tracing of yellow feather broilers in the flat breeding house: the first is the fast location of yellow feather broilers and the second is the tracking accuracy. In this paper, the foreground detection method based on colour features and YOLOv3 algorithm is used to quickly identify yellow feather broilers respectively, and then Kalman filter and Hungarian matching algorithm are used to track yellow feather broilers in the flat breeding house. The traditional algorithm has a poor recognition effect on the aggregation behaviour of broilers, resulting in poor follow-up tracking effect. Through YOLOv3 training and detection, the aggregated broilers can be well separated. The detection precision and recall rate are 98.8% and 87.5% respectively, far exceeding the accuracy and recall rate of the traditional algorithm. The model combining YOLOv3 and tracking algorithm can quickly and accurately identify and track the yellow feather broilers in the flat breeding house, which provides a new method for the detection of the movement rule and motion trail of the yellow feather broilers.

摘要

针对平养舍中黄羽鸡追踪存在的两个问题：一是黄羽鸡的快速定位，二是追踪精确度，本文使用了基于颜色特征的前景检测方法和 YOLOv3 算法分别对黄羽鸡进行快速地识别，再利用卡尔曼滤波器和匈牙利匹配算法对平养舍中的黄羽鸡进行追踪。传统算法对于鸡的聚集行为的识别效果较差，导致其后续的追踪效果不佳，通过 YOLOv3 的训练检测，可以很好的将聚集的鸡只分割出来，其检测的精确率和召回率分别为 98.8%和 87.5%，远超传统算法的精确率和召回率。使用 YOLOv3 和追踪算法相结合的模型可以快速精确的将平养舍中的黄羽鸡识别出来并进行追踪，这为黄羽鸡的运动规律和活动轨迹检测提供了新方法。

INTRODUCTION

Multiple object tracking (MOT) is a kind of important problem that has been widely concerned in machine vision. Single Object Tracking (SOT) mainly focuses on designing complex appearance models or motion patterns to solve challenging problems such as scale changes, plane rotation and illumination changes. However, multiple object tracking has two additional tasks to be solved: determine the number of targets (usually changing over time) and maintain their respective identities. In addition to the common problems of SOT and MOT, MOT also needs to deal with more complex key issues including frequent occlusion, track initialization and termination, similar appearance and interaction between multiple objects. At present, the object tracking is mainly aimed at people and there is little research on broiler tracking in flat breeding houses. In addition, due to the influence of lighting and the aggregation behaviour of broilers, it is more difficult to track multiple objects. At present, multiple object tracking algorithms are mainly divided into probability-based and decision-based optimization aspects (*Luo W., 2017*). The general mode of object tracking is shown in Figure 1.

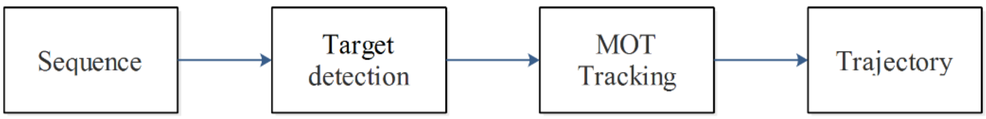


Fig. 1 - The general mode of object tracking

---

[1] *Qiyue Sun, Ms.; Tinghui Wu, Ms; Xiuguo Zou, Assoc. Prof. Ph.D.; Xinfa Qiu, Prof. Ph.D.; Heyang Yao, MAE. Stud.; Shikai Zhang, MAE. Stud.; Yuning Wei, MAE. Stud.*

Reid proposed a multi-hypothesis multiple object tracking algorithm in the research on automatic tracking of radar signals. Its essence is based on the expansion of the Kalman filter tracking algorithm in multiple object tracking problems (*Zhang L., 2008*). Nummiaro proposed a particle filter multiple object tracking algorithm based on detection reliability (*Nummiaro K., 2003*). The detection results of each frame are associated with the existing object tracks by using the greedy matching algorithm, and the particle swarm weight of each object is calculated by using the association results as the observation likelihood probability in the particle filter framework. Butt proposed a multiple object tracking algorithm based on minimum cost network flow optimization (*Butt A.A., 2013*). This algorithm is an offline multiple object tracking algorithm based on deterministic optimization. Xiang et al. proposed a multiple object tracking algorithm based on Markov decision, which is different from the previous online multiple object tracking algorithm based on the probability model. They used the Markov decision process (MDP) to derive the generation of each trajectory. This is an online object tracking algorithm based on the deterministic derivation of machine learning (*Xiang Y., 2015*). Choi proposed an approximate on-line multiple object tracking algorithm based on local flow characteristics. Besides, there are also lots of multiple object tracking algorithm based on deep learning in recent years (*Choi W., 2015*). Leal-Taixé proposed a multiple object tracking algorithm based on Siamese convolutional neural network (*Leal-Taixe L., 2016*). Tang et al. proposed multi-person tracking by multicut and deep learning (*Tang S., 2016*). Chu et al. conducted statistical analysis on the drift of tracking algorithm in pedestrian multiple object tracking problem and found that mutual occlusion is an important reason for the drift of tracking algorithm when different pedestrians interact (*Chu Q., 2017*). Therefore, an object tracking algorithm based on spatiotemporal attention mechanism learning was proposed. Kim et al. proposed a multiple object tracking algorithm based on neural gating using bilinear LSTM (*Kim C., 2018*).

In the aspect of target detection, compared with the traditional algorithm, the deep neural network has a strong ability to extract features and has high accuracy. Target detection algorithms are divided into the Two-Stage method and the One-Stage method, among which the representative of the Two-Stage method is the target detection model based on region proposal. Girshick et al. put forward the R-CNN network and Fast R-CNN network, and used a selective search method to replace the traditional sliding window (*Girshick R., 2015; Girshick R., 2016*). Girshick proposed the Faster R-CNN network, which uses the region proposal network to generate candidate frames, and then classified and coordinated regression of these candidate frames, thus greatly improving the detection accuracy (*Ren S., 2016*). These methods are called Two-Stage methods because the generation of candidate frames and prediction are divided into two steps. The network that performs these two operations at the same time is called the One-Stage method, and the representatives of the One-Stage method are SSD and YOLO (*Liu W., 2016*). In 2016, Redmon et al. proposed the YOLO network including YOLOv1, YOLOv2 and YOLOv3 (*Redmon J., 2016; Redmon J., 2017; Redmon J., 2018*). Its core idea is to use the whole map as the input of the network and directly return to the location of the bounding box and its category at the output layer. From R-CNN to Fast R-CNN, Faster R-CNN to YoloV1, YoloV2, YOLOv3, target detection algorithms are moving towards a faster and more accurate direction.

However, these algorithms are seldom applied in the identification and tracking of broilers in the flat breeding house, and because the flat breeding house is larger than the cage breeding house, the activities of broilers are freer and the broilers have aggregation behaviour, which further increases the difficulty of tracking many broilers in the flat breeding house. This paper is based on the traditional multiple object tracking algorithm and the advanced multiple object tracking algorithm based on deep learning to track the yellow feather broilers in the flat breeding house. Therefore, the activity of broilers in different periods can be judged and their activity tracks can be recorded.

MATERIALS AND METHODS

Data Acquisition

The broiler house used in this experiment was built in Jinniuhu Subdistrict, Luhe District, Nanjing City, Jiangsu Province. The broiler house has two chambers with symmetrical structures. Each chamber has a width of 1.9 meters, a length of 2.9 meters, and a total area of 5.51 m^2, 35 yellow feather broilers aged 24 days are respectively raised in each chamber. The experimental chamber is shown in Figure 2(a). The roof is sloping with a height of 2 meters on the west and 1.8 meters on the east to facilitate rainwater drainage (*Yao H. 2018*).

The video monitoring system adopts HIKVISION N1W monitoring host and hemispheric network camera. The focal length of the camera is 2.8 mm and the pixel is 2 million. The monitoring system is located at the centre above the broiler chamber for 24-hour real-time monitoring.

See Figure 2(b) for one frame of video images collected in the broiler chamber. The software used to process data in this paper includes MATLAB2014a, Pycharm2017.3, Keras2.2.4, Tensorflow- GPU1.13.1, OpenCV-Python 4.1.0.

(a) Photos of experimental broiler chambers *(b) A video frame of broiler chambers*

Fig. 2 - Data acquisition

Multiple object tracking algorithm

In this paper, based on the traditional foreground detection algorithm based on colour features and the deep learning YOLOv3 algorithm, the yellow feather broilers in the flat breeding chamber are detected and located respectively. The research plan in this paper adopts two models for comparative study: Model 1 is a combination of the foreground detection algorithm based on colour features and tracking algorithm, and Model 2 is a combination of detection algorithm based on YOLOv3 and tracking algorithm. The specific algorithm flow is shown in Figure 3.

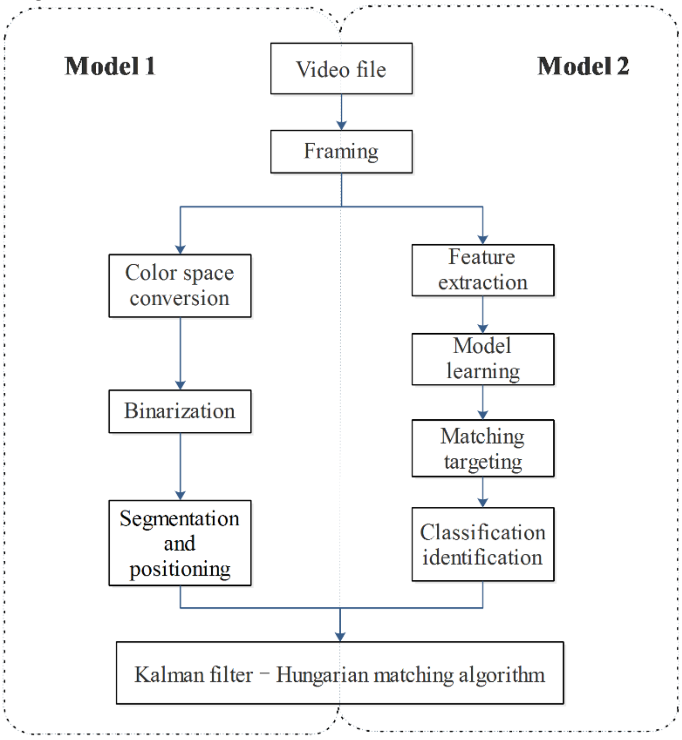


Fig. 3 - The flow chart of the comparison between Model 1 (traditional image processing algorithm) and Model 2 (YOLOv3 deep learning algorithm)

Foreground detection based on colour features

- **Colour space conversion**

The R, G and B components of RGB colour space are greatly affected by illumination, and the correlation between the three components and colour is high. It is difficult to determine the position of yellow feather broiler only by these components, so other components of the colour space are needed. In this paper, HSV colour space is selected, and the equation for converting from RGB colour space to HSV colour space is as follows (*Wang J., 2010*).

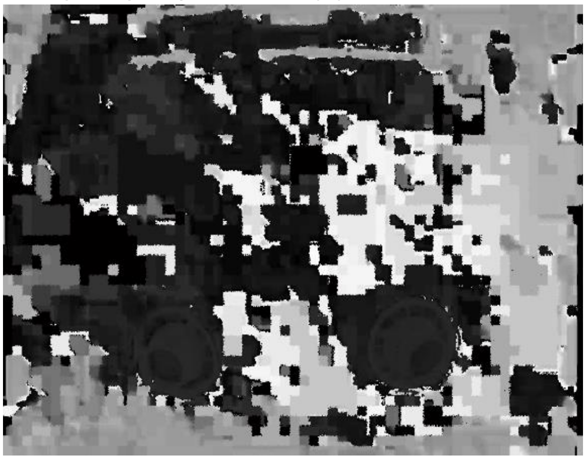
$$V=\frac{1}{3}(R+G+B) \tag{1}$$

$$S=1-\frac{3}{(R+G+B)}[min(R,G,B)] \tag{2}$$

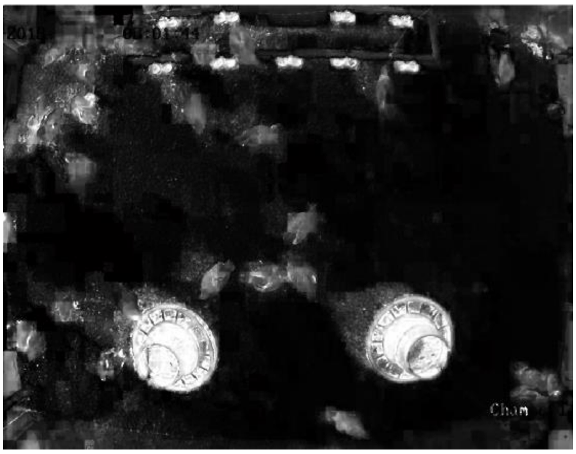
$$H=arctan[\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}] \tag{3}$$

- **Binarization**

In this paper, the threshold is determined according to the HSV colour space component image to extract the yellow feather broiler. From the HSV component image of Figure 4, it is found that the S component of RGB colour space has better distinguishing characteristics, and then the corresponding threshold interval is set in combination with other components, so that the gray value in the interval becomes 1 and the gray value not in the interval becomes 0; thus, the image is converted into a binary image, obtaining the prospect, namely yellow feather broiler, as shown in Figure 5.



(a) H component



(b) S component

(c) V component

Fig. 4 - HSV component diagram

Fig. 5 - HSV segmentation of binary image

- **Segmentation and positioning**

First, the binarized image is subjected to correlation processing, including morphological filtering to remove redundant fine spots, and filling processing to fill the morphology of the yellow feather broiler, and then a binary image of the yellow feather broiler region can be obtained, as shown in Figure 6. Therefore, the position information of each broiler is obtained by image segmentation, and the detection result is shown in Figure 7.

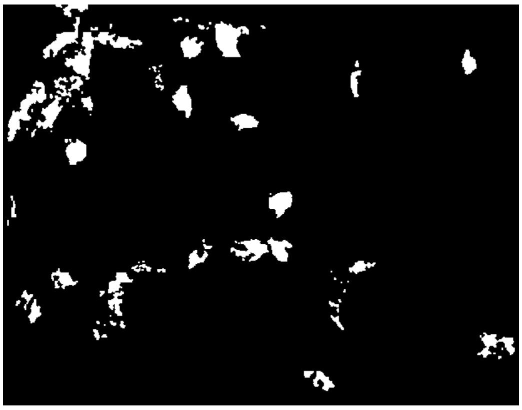


Fig. 6 - The binarized image by morphological filtering

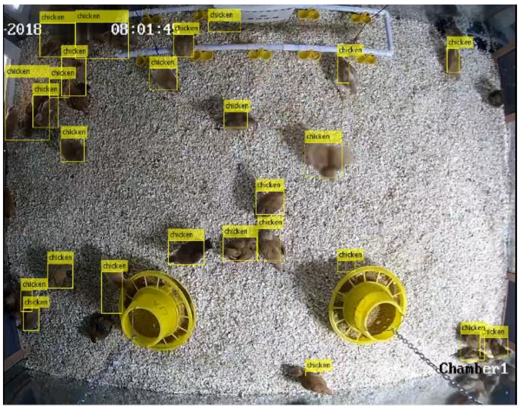


Fig. 7 - The result of foreground detection

Detection and positioning based on YOLOv3

- **Production of the training set**

The original author of YOLOv3 trained 80 species, but there is no yellow feather broiler in it so this paper will train the author's own dataset. For the problems studied in this paper, LabelImg is used to frame the targets and generate training set files, as shown in Figure 8.

Fig. 8 - Use LabelImg box to select yellow feather broiler

Attention should be paid to selecting the whole frame of the broiler as accurately as possible to ensure the accuracy of the training data set. In this paper, 50 images are selected, 35 yellow feather broilers are in each frame, totalling 1750 yellow feather broiler positioning and classification, so the sample number in the training set is 1750.

- **YOLOv3 detection**

YOLOv3 has attracted much attention due to its strong real-time detection and high accuracy. YOLOv3 is Redmon's improved target detection algorithm based on YOLOv2. Different from the network structure Darknet19 used by YOLOv2, YOLOv3 adopts a completely new network structure. As shown in Figure 9, it contains a large number of 3×3 and 1×1 convolution layers with a total of 53 convolution layers. And there are no pooling layer and full connection layer in the whole v3 structure. In the forward propagation process, the size transformation of the tensor is realized by changing the step size of the convolution kernel. YOLOv3 uses the idea of FPN (feature pyramid networks) for reference and makes predictions on three different scales, namely 13×13, 26×26 and 52×52. Each scale predicts three boxes (*Gao Q., 2017*). For a picture, if it is initially divided into K×K grids and C categories need to be predicted, then the tensor obtained for each scale is K×K[3×(4+1+C)], which includes 4 offset coordinates of the target border and the confidence score. Because the feature maps of the first two layers are fused, the model can acquire more image semantic information of the lower layer and the higher layer (*Li Y., 2019*).

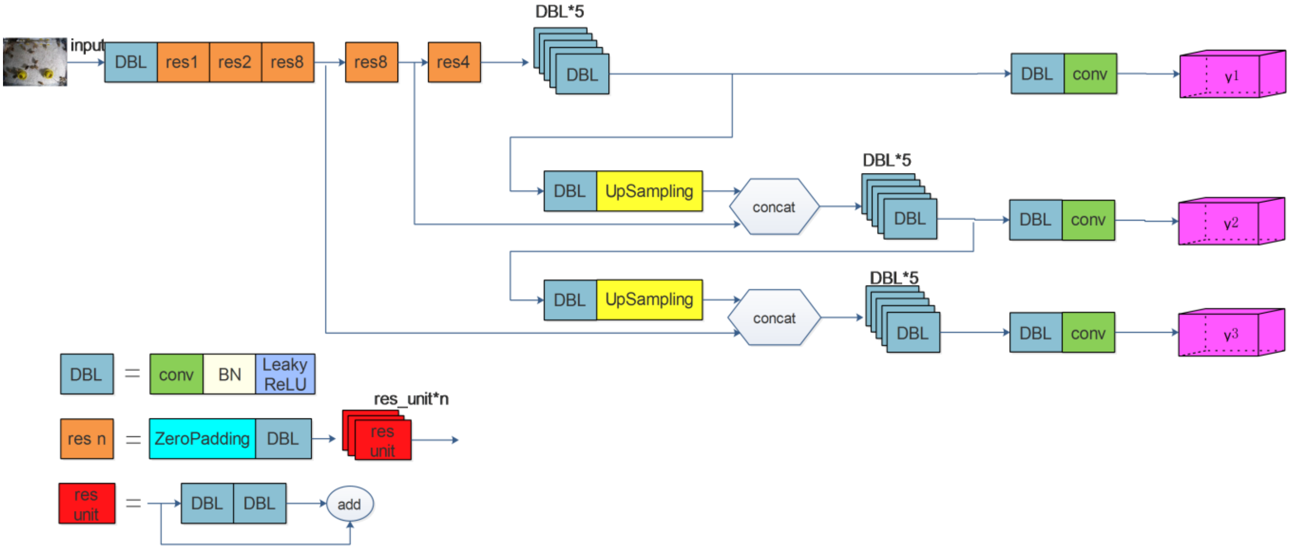


Fig. 9 - YOLOv3 network structure with the image of the broilers inputting from the DBL on the top left. DBL is a function with the name DarknetConv2D_BN_Leaky

Based on Keras and TensorFlow, this paper uses the YOLOv3 algorithm. YOLOv3 can locate and detect broilers in the video through the training of data sets, and the detection results are shown in Figure 10. The red box in the figure shows the detected objects and their categories, and the right side shows the confidence level of their detection.

Fig. 10 - YOLOv3 detection and positioning results

- **Tracking algorithm**

This paper uses a tracking algorithm that combines the classical Kalman filter and Hungarian matching algorithm. The algorithm has fast speed and excellent effect. The steps of the tracking algorithm are shown in Figure 11.

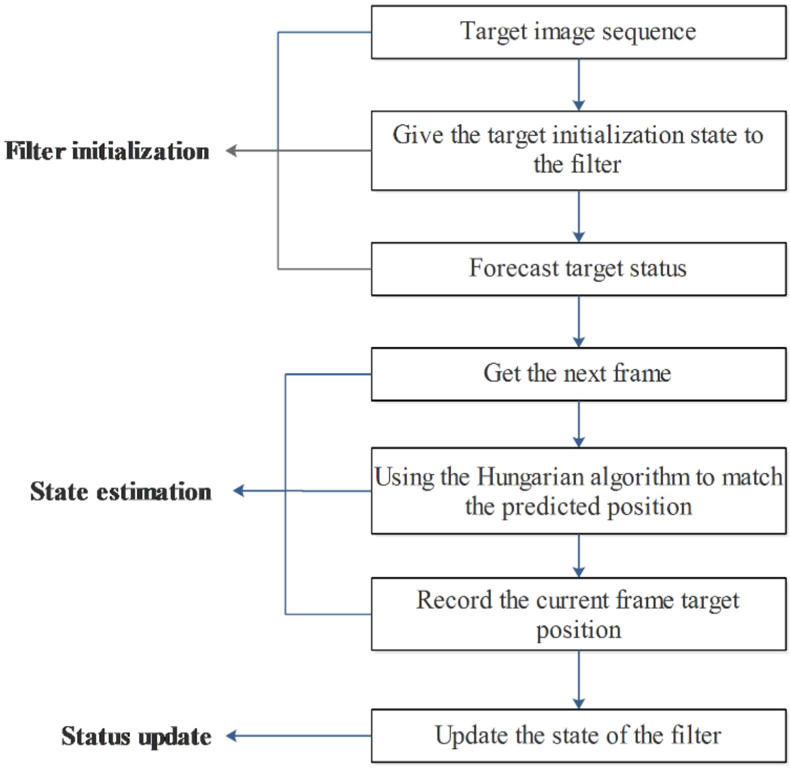


Fig. 11 - The flow chart of the tracking algorithm steps combines the classical Kalman filter and Hungarian matching algorithm

- **Kalman filtering**

Kalman filter has been widely used in many different object tracking processes. Its main advantage is that it can provide an informed guess about the future location of any given object in a dynamic environment (*Hamuda E., 2018*). Kalman is a linear estimation algorithm, which can establish the relationship between inter prediction frames. Tracking is divided into 5 states: 1) a new object appears; 2) the object matching; 3) the object occlusion; 4) the object separation; 5) the object disappears.

The positioning state $x_k$is a vector, which can include speed, etc. in addition to coordinates. The Kalman filter is used to track the moving target. The steps are divided into the following five steps.

Step 1: The current state is predicted from the state at the previous moment and the external input.

$$x_k^- = Ax_{k-1} + Bu_{k-1} \quad (4)$$

Step 2: The prediction process adds a new uncertainty Q and the previous uncertainty.

$$P_k^- = AP_{k-1}A^T + Q \quad (5)$$

Step 3: The Kalman gain (weight) is calculated from the uncertainty of the prediction result P_k^- and the uncertainty of the observation result R.

$$K_k = P_k^- H^T \left(HP_k^- H^T + R\right)^{-1} \quad (6)$$

Step 4: The prediction results and the observation results are weighted averages to obtain the state estimation at the current time.

$$x_k = x_k^- + K_k(z_k - Hx_k^-) \quad (7)$$

Step 5: Update P_k, representing the uncertainty of this state estimation.

$$P_k = P_k^- - K_k HP_k^- \quad (8)$$

Step 1 and step 2 are the processes of estimating the state of the current time based on the state of the previous time, i.e. the prediction. Steps 3, 4 and 5 are the process of synthesizing the estimated state and the observed state of the current time to estimate the optimal state, i.e. the correction. See Table 1 for the meaning in the equation.

Table 1

The descriptions of equation symbols

Symbol	Meaning
x_k^-	Predicted value at time K
x_k	Status at time K
u_k	The effect of the outside on the system at time K
z_k	Observations at time K
K_k	Kalman gain at time K
A	State transition matrix （related to a specific linear system）
B	Input control matrix
P	Error matrix
Q	Predicted noise covariance matrix
R	Measurement noise covariance matrix
H	Observation matrix

- **Hungarian matching algorithm**

The matching uses the Hungarian algorithm; here the moving object detected in the new frame is matched to the corresponding track. The matching process is realized by minimizing the sum of Euclidean distances between the centroid predicted by Kalman and the detected centroid, which can be divided into the following two steps.

Step1：Calculating a loss matrix with a size of [*M N*], wherein *M* is the number of tracks and *N* is the number of moving objects detected.

Step2：Solving loss matrix.

RESULTS

Experimental results

- **Comparison of detection results**

Model 1 uses a foreground detection method based on colour features and the detection results are shown in Figure 12. Model 2 uses the object detection and positioning algorithm based on YOLOv3, and the detection result is shown in Figure 13.

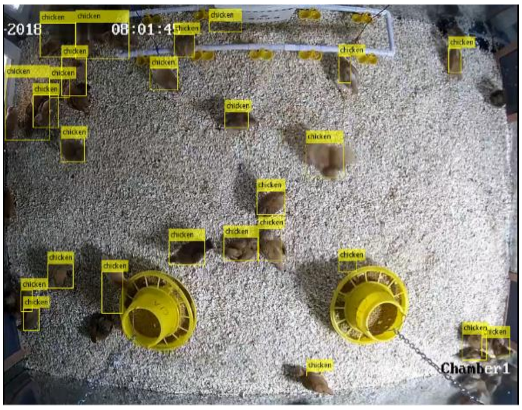


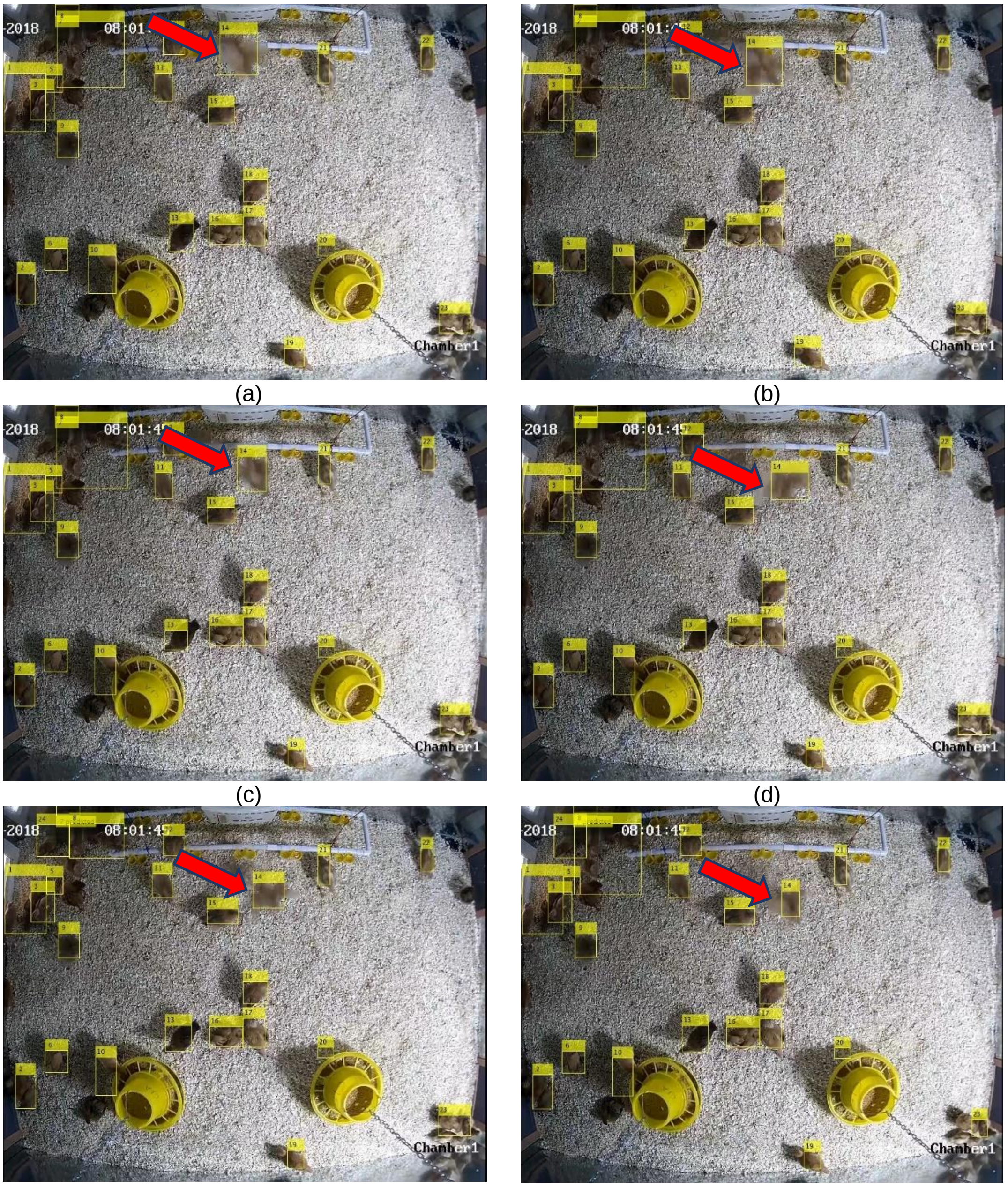
Fig. 12 - The tracking results of Model 1

Fig. 13 - The tracking results of Model 2

It can be clearly seen from the two detection results that 21 detection boxes appear in Model 1, of which only 16 are the correct targets for detection, and the remaining 5 have the situation that other objects are identified as yellow feather broilers or multiple yellow feather broilers are identified as one yellow feather broiler. There are also 21 detection boxes in Model 2, but each of them can effectively locate yellow feather broilers.

- **Comparison of tracking results**

Model 1 uses a combination of colour feature-based foreground detection and tracking algorithm to track yellow feather broilers in a flat breeding chamber. The tracking results are shown in Figure 14. Model 2 uses YOLOv3-based target detection and location algorithm combined with the tracking algorithm to track yellow feather broilers in the flat breeding chamber; the tracking results are shown in Figure 15.



(a) (b)

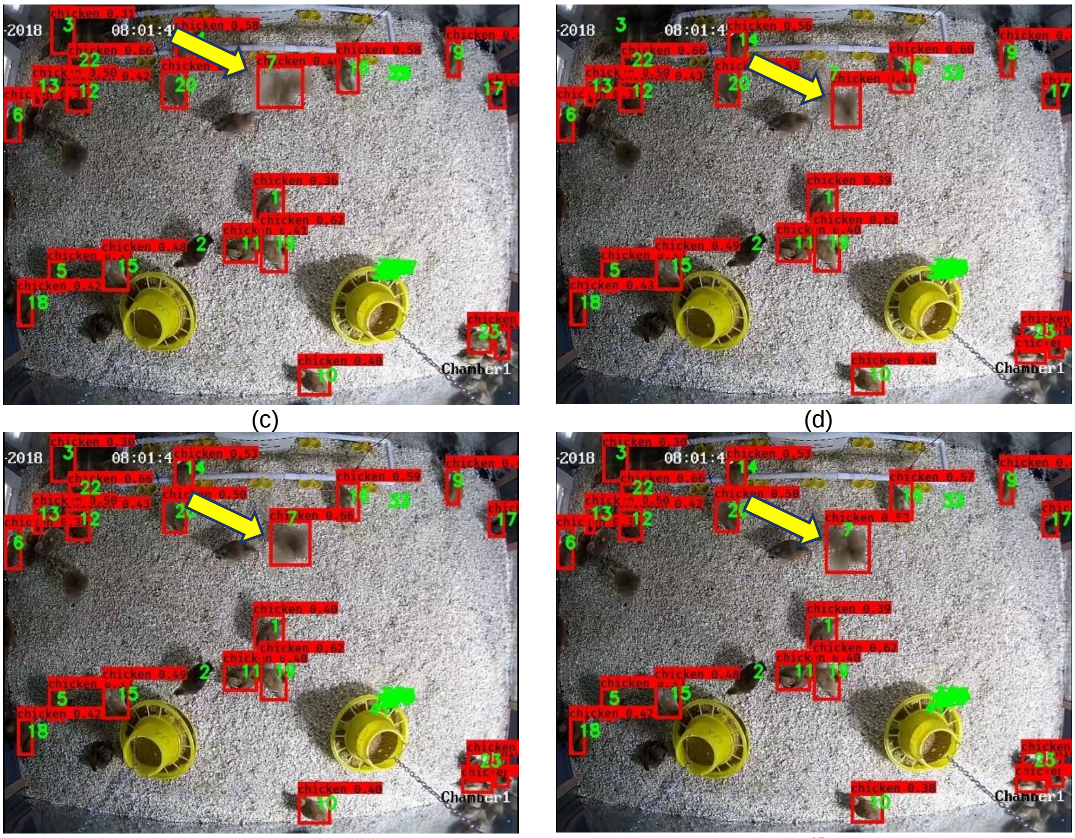
(c) (d)

(e) (f)

Fig. 14 - The tracking results of Model 1 from (a) to (f)

(a) (b)

Fig. 15 - The tracking results of Model 2 from (a) to (f)



(c) (d)

(e) (f)

Fig. 15 - The tracking results of Model 2 from (a) to (f)

The ID of each yellow feather broiler will be displayed in the upper left corner of the calibration box in Figure 14, and the target box will change continuously with the movement of the yellow feather broiler. The yellow feather broiler indicated by the red arrow in the figure can be tracked all the time during the movement and its ID (ID=14) can be kept unchanged. The green number in Figure 15 shows the ID of each yellow feather broiler. The target box will also follow the continuous changes of the yellow feather broiler's movement. The yellow feather broiler indicated by the yellow arrow in the figure will keep its ID (ID=7) unchanged during the movement. It can be seen from this that both models can effectively track the yellow feather broilers in the flat breeding chamber.

Detection evaluation

The evaluation metrics of the MOT method are very important because they provide a way for fair and quantitative comparison. MOT evaluation metrics are roughly divided into two groups, which are used for evaluation, detection and tracking (*Luo W., 2017*). Because the detection algorithms used in this paper are different, the comparative evaluation metrics of this model can be divided into the following aspects from the perspective of detection.

Precision: the proportion of the number of yellow-feather broilers identified as yellow feather broilers to the total identification number. The definition is shown in Equation (9).

$$Precision=\frac{TP}{TP+FP} \tag{9}$$

Recall: the proportion of the number of broilers identified as broilers to all broilers. The definition is shown in Equation (10).

$$Recall=\frac{TP}{TP+FN} \tag{10}$$

TP refers to the number of yellow feather broilers identified as yellow feather broilers by the detection model, FP refers to the number of none yellow feather broilers identified as yellow feather broilers, FN refers to the number of yellow feather broilers identified as none yellow feather broilers (*Liu B., 2019*). The comparison results of two models are shown in Table 2.

Table 2

The comparison results of the two models

Evaluation metrics	Model 1	Model 2
Precision / %	87.5	98.8
Recall / %	70.0	87.5

Follow-up evaluation

The tracking metrics used in this paper include the multiple object tracking accuracy (MOTA), missed detection rate ($\overline{m}$), misjudgement rate ($\overline{fp}$), mismatching rate ($\overline{mme}$) (*Bernardin K., 2008*). MOTA metric is used to measure the overall tracking accuracy and is the most representative.

The multiple object tracking accuracy rate (MOTA) is defined in Equation (11).

$$MOTA=1-\frac{\sum_t (m_t+fp_t+mme_t)}{\sum_t g_t} \tag{11}$$

where: m_t, f_{pt}, and mme_t represent the number of misses, the number of false positives and the number of mismatching, respectively. g_t represents the number of the true object at time t. For time t, MOTA can be seen as the result of the three error ratios in Equation (12), Equation (13) and Equation (14).

(1) Missed detection rate:

$$\overline{m}=\frac{\sum_t m_t}{\sum_t g_t} \tag{12}$$

(2) Misjudgement rate:

$$\overline{fp}=\frac{\sum_t fp_t}{\sum_t g_t} \tag{13}$$

(3) Mismatching Rate:

$$\overline{mme}=\frac{\sum_t mme_t}{\sum_t g_t} \tag{14}$$

The MOTA evaluation results adopted in models 1 and 2 are shown in Table 3. As it can be seen from Table 3, although model 1 and model 2 use the same multiple object tracking algorithm, the tracking effect of model 1 is poor due to its poor detection effect.

Table 3

The comparison results of the two models

Evaluation metrics	Model 1	Model 2
MOTA / %	59.7	78
$\overline{m}$ / %	19.8	13.2
$\overline{fp}$ / %	12.5	1.2
$\overline{mme}$ / %	8.0	7.6

CONCLUSIONS

In this paper, Model 1 (traditional image processing algorithm) and Model 2 (YOLOv3 deep learning algorithm) are used to compare the tracking of broilers in the flat breeding chamber and the following conclusions are obtained.

(1) Model 1 cannot solve the problems that broilers cannot be segmented when they are gathered and overlapped and it is difficult to identify broilers which are small or on the edge of the video. In Model 2, gathered broilers, edge broilers and smaller broilers can be well segmented and identified. The detection accuracy and recall rate of Model 2 are 98.8% and 87.5% respectively, which is much higher than that of Model 1.

(2) Both Model 1 and Model 2 can track the target effectively, but Model 1 has a lower tracking accuracy than Model 2 due to poor detection effect.

(3) Model 2 can effectively identify broilers and track them. The next step is to embed the model into the intelligent hardware system, thus further promoting the application of vision technology in livestock breeding.

ACKNOWLEDGEMENT

This research was jointly supported by the Fundamental Research Funds for the Central Universities of China (KYTZ201661), China Postdoctoral Science Foundation (2015M571782), Jiangsu Agricultural Machinery Foundation (GXZ14002) and University Student Entrepreneurship Practice Program of Jiangsu Province (NO. 201810307010P).

REFERENCES

[1] Bernardin K., Stiefelhagen R., (2008), Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics, *Journal on Image and Video Processing*, Vol.2008, pp.1-10, London/England;

[2] Butt A., Collins R.T., (2013), Multi-target Tracking by Lagrangian Relaxation to Min-cost Network Flow, *CVPR*, pp.1846-1853, Portland/Oregon;

[3] Choi W., (2015), Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor, *ICCV*, pp.3029-3037, Santiago/Chile;

[4] Chu Q., Ouyang W., Li H., et al., (2017), Online Multi-Object Tracking Using CNN-based Single Object Tracker with Spatial-Temporal Attention Mechanism, *ICCV*, pp.4846-4855, Venice/Italy;

[5] Gao Q., Fang H., (2017), HOG Pedestrian Detection Algorithm Based on Multi-Convolution Feature Fusion, *Computer Science*, Vol.44, pp.199-201+232, Chongqing/China;

[6] Girshick R., (2015), Fast R-CNN, *ICCV*, pp. 1440-1448, Santiago/Chile;

[7] Girshick R., Donahue J., Darrell T., et al., (2016), Region-based Convolutional Networks for Accurate Object Detection and Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.38, Issue1, pp.142-158, Los Angeles/USA;

[8] Hamuda E., Ginley B.M., Glavin M., et al., (2018), Improved Image Processing-Based Crop Detection Using Kalman Filtering and the Hungarian Algorithm, *Computers and Electronics in Agriculture*, Vol.148, pp.37-44, Oxford/England;

[9] Kim C., Li F., Rehg J.M., (2018), Multi-object Tracking with Neural Gating Using Bilinear LSTM, *ECCV*, Vol.11212, pp.208-224, Munich/Germany;

[10] Leal-Taixe L., Ferrer C., Schindler K., (2016), Learning by Tracking: Siamese CNN for Robust Target Association, *CVPRW*, pp.418-425, Las Vegas/USA;

[11] Li Y., Hou L., Wang C., (2019), Moving target detection in automatic driving based on YOLOv3, *Computer Engineering and Design*, Vol.40, pp. 1140-1144, Beijing/China;

[12] Liu B., Wang S., Zhao J., et al., (2019), Ship Tracking Recognition Based on Darknet Network and YOLOv3 Algorithm, *Journal of Computer Applications*, Vol.1001-9081, pp.1-7, Sichuan/China;

[13] Liu W., Anguelov D., Erhan D., et al., (2016), SSD: Single Shot MultiBox Detector, *ECCV*, Vol.9905, pp.21-37, Amsterdam/ Netherlands;

[14] Luo W., Xing J., Milan A., et al., (2017), Multiple Object Tracking: A Literature Review, *CVPR*, Hawaii/USA;

[15] Nummiaro K., Koller-Meier E., Gool L.V., (2003), An Adaptive Color-Based Particle Filter, *Image and Vision Computing*, Vol.21, Issue 1, pp.99-110, Amsterdam/Netherlands;

[16] Redmon J., Divvala S., Girshick R., et al., (2016), You Only Look Once:Unified, Real-Time Object Detection, *CVPR*, pp.779-788, Las Vegas/USA;

[17] Redmon J., Farhadi A., (2017), YOLO9000 Better, Faster, Stronger, *CVPR*, pp.6517-6525, Hawaii/USA;

[18] Redmon J., Farhadi A., (2018), YOLOv3: An incremental improvement, *CVPR*, Vol.1804.02767, Salt Lake City/USA;

[19] Ren S., He K., Girshick R., et al., (2016), Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.39, pp.1137-1149, Los Angeles/USA;

[20] Tang S., Andres B., Andriluka M., et al., (2016), Multi-person Tracking by Multicut and Deep Matching, *ECCV 2016 Workshops*, Vol.9914, pp.100-111, Amsterdam/Netherlands;

[21] Xiang Y., Alahi A., Savarese S., (2015), Learning to Track: Online Multi-Object Tracking by Decision Making, *ICCV*, pp.4705-4713, Santiago/Chile;

[22] Yao H., Sun Q., Zou X., et al., (2018), Research of yellow-feather chicken breeding model based on small chicken chamber, *INMATEH-Agricultural Engineering*, Vol.56, Issue 42, pp.91-100, Bucharest/Romania;

[23] Wang J., (2010), Research on color image segmentation algorithm based on color space segmentation, *Qufu Normal University*, M.S., Qufu/China;

[24] Zhang L., Li Y., Nevatia R., (2008), Global Data Association for Multi-Object Tracking Using Network Flows, *CVPR*, pp.1-8, Alaska/USA.