Scientific Reports

https://doi.org/10.1038/s41598-026-43523-7

Article in Press

Lightweight multiscale behavior recognition for caged laying hens using an enhanced YOLOv8 framework

Received: 5 December 2025

Accepted: 4 March 2026

Published online: 25 March 2026

Cite this article as: Tang Y., Wei J., Xie B. *et al.* Lightweight multiscale behavior recognition for caged laying hens using an enhanced YOLOv8 framework. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-43523-7

Yurong Tang, JingGe Wei, Binbin Xie, Rui Kang, Chao Yuan, Jing Liu, Zhichao Mo, Longshen Liu & Mingxia Shen

Lightweight Multiscale Behavior Recognition for Caged Laying Hens Using an Enhanced YOLOv8 Framework

Yurong Tang [1, 2]; JingGe Wei [1, 3];Binbin Xie [1, 3];Rui Kang [1, 3]; Chao Yuan [1, 3]; Jing Liu [1, 3]; Zhichao Mo [1, 3]; Longshen Liu [1, 3];Mingxia Shen [1, 3, *]

(1. Key Laboratory of Breeding Equipment Ministry of Agriculture and Rural Affairs of China, Nanjing China, 210031; 2.College of Engineering, Nanjing Agricultural University, Nanjing China 210031; 3. School of Artificial Intelligence, Nanjing Agricultural University, Nanjing China 210031)
Supported by:National Key R&D Program of China（2023YFD2000800）
* Correspondence: Mingxia Shen, Professor, PhD, Mainly Engaged in Research on Smart Animal Husbandry (mingxia@njau.edu.cn)

Abstract

This study presents a lightweight deep-learning framework for recognizing key health-related behaviors of caged laying hens, addressing the challenges of dense housing, frequent occlusions, and subtle action differences. Building upon YOLOv8n (serving as our base model), we introduce three major enhancements: (1) a C2f-FasterNet-EMA backbone that improves multiscale feature extraction; (2) a FasterNet-based neck combined with the Dysample upsampler to refine small-object localization while reducing computational cost; (3) an EMASlideLoss function that alleviates sample imbalance and stabilizes the training process. Evaluated on the Lukou Dataset, which includes four target behaviors (eating, open-mouth breathing, self-pecking, and mutual pecking), the improved model achieves mAP@50 scores of 98.15%, 81.03%, 93.65%, and 94.32% for each behavior, respectively. Overall, compared with the retrained baseline YOLOv8n under identical experimental settings, the proposed method attains a 2.26% improvement in overall mAP@50 while reducing the model size by 22.92% relative to the baseline.

Keywords: Caged laying hens; Behavior detection; Improve YOLOv8n; Smart farming; Intelligent chicken farming equipment

1.Introduction

The egg laying industry plays an important role in the revitalization of rural areas in China[1]. The intensive feeding mode has the advantage of low cost and high efficiency, but it is prone to frequent outbreaks of diseases, leading to a decline in egg quality[2-3]. With the increasing demand for high-quality animal products, timely detection and prevention of abnormal conditions in laying hens is an effective means to achieve high-quality production without resistance. Studies have shown that the behavior information of laying hens, including their behavior, is closely related to their physiological status. By obtaining behavioral data, the health status of laying hens can be evaluated in a timely manner[4].

In large-scale operations, analyzing poultry behavior can help farm managers oversee flocks more precisely [5]. Monitoring behaviors such as feeding, drinking, fighting, and pecking assists in assessing hen welfare [6–7]. Current multimodal monitoring technologies primarily focus on activities like feeding and lameness [8]. Beyond unimodal visual models, recent works (e.g., ARTEMIS [9]; MSQNet [10]) integrate visual, textual, and even audio modalities

to enhance recognition robustness, especially for complex or ambiguous behaviors. The release of large-scale, diverse datasets like the Animal Kingdom Dataset [10] has been transformative, providing standardized evaluation for multilabel action recognition across hundreds of species. This has shifted the field from species-specific, small-scale studies to generalized models that can adapt to multiple animals—including poultry—with minimal fine-tuning. Further analysis of these behaviors is essential for early detection of health issues, optimizing management strategies, and improving farming efficiency [11-12].

However, there are still some research gaps[13-14]. As summarized in the recent survey by Edoardo Fazzari [15], the field of deep learning-based animal behavior analysis still faces several unresolved limitations: (1) the lack of large-scale, standardized datasets covering diverse poultry behaviors (especially rare or subtle actions like early lameness signs); (2) existing models often prioritize accuracy over efficiency, making them unsuitable for real-time monitoring on resource-constrained farm devices; and (3) few approaches effectively generalize across different chicken breeds, ages, or farm environments. While prior works [13-14] touch on species-specific gaps, the survey [15] underscores these as universal challenges that motivate our research.

In addition, there are still three challenges in terms of the overall research background. Firstly, most studies have limited coverage of behaviors that sensitively reflect health issues [16]. According to Professor Yao Wen from the School of Animal Sciences at Nanjing Agricultural University, certain behaviors of chickens are associated with certain early diseases or sub-health symptoms[17], such as opening the mouth and abnormal pecking. Secondly, there are few existing methods based on cage based motion detection mode [18]. In addition, most studies have focused on flat agricultural models, which limits their practical applicability in different agricultural environments [19].

Although many studies have attempted to quantify poultry behavior, most current monitoring methods are constrained by limited temporal precision or narrow scene adaptability. These limitations stem from difficulties in tracking small-sized hens accurately and overcoming environmental interference in complex farming systems. Therefore, there is a critical need for efficient behavior recognition methods suitable for group-housing environments that can monitor multiple behaviors in real time under commercial conditions to effectively assess hen health and welfare.

In view of this, our research aims to classify the typical behaviors of captive laying hens and propose an improved object detection algorithm based on YOLOv8 to accurately identify behaviors, locate individuals, and count occurrence times. Meanwhile, this work adopts EMA (Efficient Multi-Scale Attention) [20-22] to enhance the model's ability to capture fine-grained and multi-scale features of poultry behaviors—an attention module that has been previously integrated into YOLOv8 [20] and YOLOv9 [21] for improved object detection and action recognition across diverse scales. This method will promote effective health and welfare assessments while supporting intelligent poultry farming management.

The main contributions are as follows:

(1) A behavior monitoring platform designed for caged poultry farming environments was developed. The system incorporates a robust video acquisition setup, including surveillance cameras, Jetson Nano modules[23], and durable lithium-ion batteries, enabling stable capture of high-resolution hen behavior footage. Through optimized camera placement and parameter configuration, obstructions such as feeding troughs are minimized, ensuring reliable data collection. Additionally, the platform includes a standardized data processing pipeline involving frame extraction, image screening, expert annotation, and dataset partitioning. This infrastructure supports the construction of a specialized behavior database, forming the basis for accurate ethological analysis.

(2) The YOLO(You Only Look Once) architecture was enhanced through four key innovations: the FastNet module, which uses partial convolutions to reduce computational redundancy and memory access; the EMA efficient multi-scale attention mechanism, improving spatial feature extraction via multi-scale focus; the Dysample upsampler, enabling resource-efficient resolution enhancement; and the EMASlideLoss function, which mitigates class

imbalance using triple-accelerated strategies.

These modifications improve detection robustness and reduce missed detections. In comparative experiments with mainstream detectors, the proposed method demonstrated superior precision and inference efficiency in recognizing avian behaviors. This edge-deployable solution shows significant potential for mobile inspection robotics, advancing intelligent poultry management toward precision farming paradigms.

2. Materials and Methods

The study involved non-invasive observation only and was conducted with the consent of the farm owner. Live chickens were used for image collection, but all image acquisition was performed using a non-invasive, non-contact visual system. No physical contact with the animals was involved at any stage of the image collection process. The study was conducted in full compliance with ethical guidelines and regulations, and permission for the study was obtained from the farm owner. The farm environment was carefully selected to avoid any distress to the animals.

The animal experiments were approved by the Institutional Animal Care and UseCommittee of Nanjing Agricultural University according to the Guidelines on EthicalTreatment of Experimental Animals (2006) No. 398 set by the Ministry of Science andTechnology (2006, Beijing, China) (NJAU.No20241213260).

2.1 Optical Inspection System

This section describes the hardware architecture used for data acquisition in this study.

The research was conducted at Lukou Poultry Industry Co., Ltd. in Nanjing, P.R. China from October to December 2023. The chicken cages in this large-scale breeding farm are stacked in four tiers, each holding eight hens. The experimental subjects were "HY-LINE VARIETY BROWN" laying hens during their peak egg-laying period.

The data collection system was deployed across four poultry houses. The video capture system comprises three main components: surveillance cameras, Jetson Nano developer kits, and outdoor lithium battery packs.

2.1.1 Surveillance Cameras

Each layer of laying hens has a dedicated camera. This experiment selected the Hikvision camera model "DS-2CD2347SF (D) WDV3-LS" with a focal length of 3.0 mm, no fill light, built-in backlight compensation, strong light suppression, and 3D digital noise reduction, suitable for digital image acquisition in cage chicken coops.To minimize the obstruction of the feeding trough on the view of the chickens, the cameras are set at an oblique upward shooting angle. The size of a single chicken coop is 60 × 40cm. According to this focal length, the overhead shooting height for the fourth layer is set to 60cm. The cameras output color video, which provides rich visual information for subsequent analysis. The video recording is carried out from 6:00 to 18:00 every day for 20 consecutive days, ensuring a sufficient amount of data is collected.

2.1.2 Jetson Nano Development Boards

Each USB(Universal Serial Bus) camera is connected to a Jetson Nano development board. This board serves as a data processing and storage center. It has the computing power to handle the incoming video streams from the cameras and save the collected chicken video data in the ".mp4" format.

2.1.3 Outdoor Lithium Batteries

The Jetson Nano development boards are powered by outdoor lithium battery packs. This power supply configuration ensures system mobility and enables continuous operation without being constrained by fixed power infrastructure . The captured video data are stored directly on the Jetson Nano

boards before being transferred to portable hard drives for off-line analysis. This approach facilitates subsequent data processing and in-depth investigation into key behaviors of caged laying hens based on video recordings. Each tier of hens is monitored by a dedicated camera.This article presents the hardware architecture diagram, as illustrated in **Figure 1**.

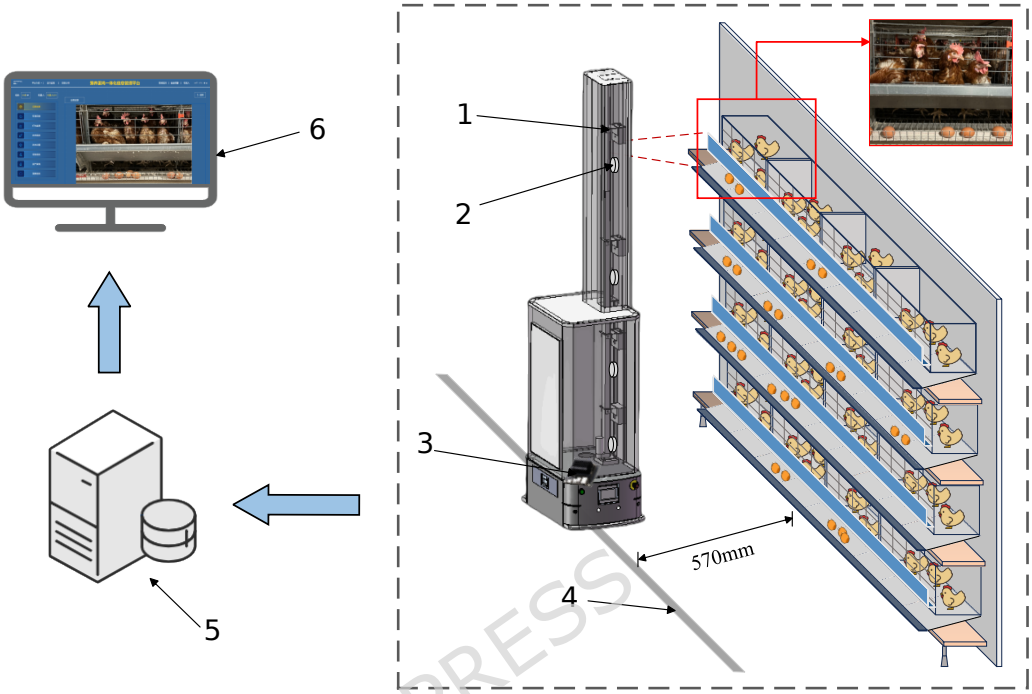


Figure.1 Image data acquisition hardware platform

(1.Industrial cameras; 2.Environmental sensors; 3.Jetson Nano development module; 4.Magnetic stripe orbits; 5. Local servers; 6.Cloud based Big Data Platforms)

2.2 Dataset Preparation

2.2.1 Data Preprocessing

To ensure dataset quality and mitigate overfitting, the following preprocessing steps were performed:

Video Screening and Frame Extraction: Raw videos collected from Lukou Poultry Industry Co., Ltd in Nanjing, P.R. China. were filtered to remove segments containing human interference or chicken stress responses, retaining 2,832 raw frames. Python scripts extracted 1 frame per second, resulting in 2,035 valid images after removing blurry, occluded, or poorly lit samples.

Data Cleaning: For data cleaning, we manually inspected and excluded frames with severe occlusion (>60% body coverage), low resolution (<720p), motion blur (laplacian variance < 150), and poor lighting (average brightness <200 lux or >1200 lux), ensuring only representative, high-quality samples were retained for model training.

To more intuitively quantify the specific characteristics of frame discard during the preprocessing stage, including the distribution of discard rates and their key contributing factors across different deployment environments, Table 1 is supplemented to present the statistical results of automated frame discard in poultry farming scenarios. This data is derived from the same experimental settings as those for raw video acquisition—specifically the upper cage tiers (3rd and 4th) and lower cage tiers (1st and 2nd) inside the poultry house. Compiled from 30-minute deployment windows (totaling 15 FPS × 1800 sec = 27,000 frames) during 2-hour continuous operation, the data clearly reflects the impact of varying light conditions on frame quality as well as the screening performance of the preprocessing workflow.

Table 1 Automated Frame Discard Rates Over 30-Minute Deployment Windows

Deployment Environment	Total Frames Captured (30 mins)	Discarded Frames	Discard Rate	Reason for Discard (Top 2)
lower tiers (1st & 2nd)	2,700	528	19.55%	Occlusion (12.3%), Blur (7.25%)
upper tiers (3rd & 4th)	2,700	672	24.89%	Poor Lighting (14.2%), Blur (10.69%)

2.2.2 Behavioral Annotation

A total of 2,035 images were annotated using the open-source tool LabelImg[24], identifying four key behaviors of caged laying hens: one normal and three abnormal (Table 2). Rectangular bounding boxes were used to delineate the head regions and classify the behaviors into four categories: “eat” (normal feeding), “open_mouth” (abnormal breathing), “peck” (self-feather pecking), and “peck_other” (mutual feather pecking). All annotations were stored in XML (Extensible Markup Language) format, which includes metadata such as image dimensions, bounding box coordinates, and class labels (Figure 2).

2.2.3 Dataset Splitting

To ensure robust model training, reliable hyperparameter optimization, and unbiased evaluation of generalization performance, the final curated dataset was partitioned into training, validation, and test subsets following a standard 8:1:1 ratio—a widely adopted split in computer vision and animal behavior recognition research [9, 20,21]. This ratio balances sufficient data for model learning (via the training set) with rigorous validation of model stability (via the validation set) and objective assessment of real-world applicability (via the test set). Specifically, the training set comprises 1,628 images, which were used to optimize core model parameters and enable the network to learn discriminative features of poultry behaviors (e.g., foraging, preening, and locomotion). The validation set (204 images) was employed during the training loop to tune critical hyperparameters (e.g., learning rate, batch size) and monitor for overfitting—with training halted early if validation loss plateaued for three consecutive epochs. Finally, the test set (203 images) consisted of unseen data that was completely isolated from the training and validation processes, ensuring an unbiased evaluation of the model’s ability to generalize to novel poultry postures, environmental variations, and behavioral instances not encountered during training. To avoid data leakage and maintain the integrity of the evaluation, the splitting process was performed randomly while preserving the class distribution of behaviors across all three subsets (e.g., the proportion of “lameness” or “foraging” samples remained consistent with the full dataset).

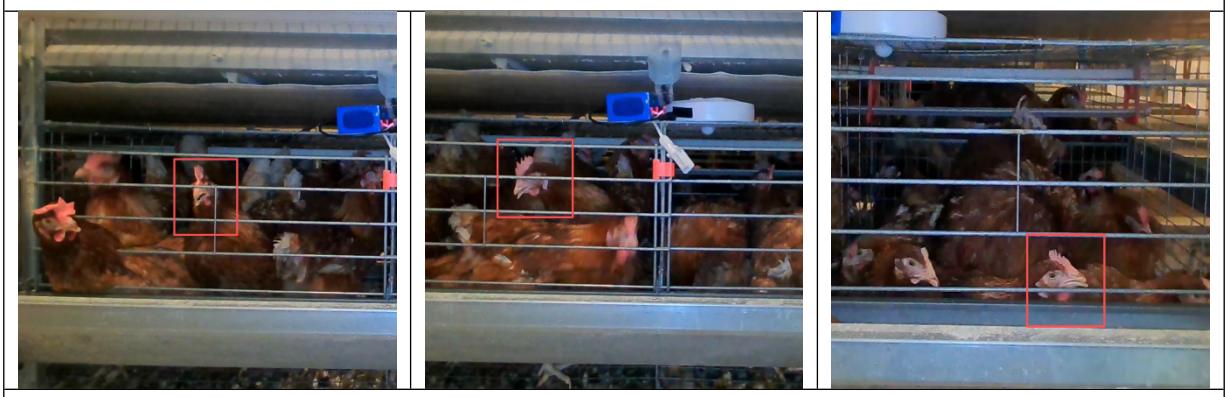
The health status of chickens is closely linked to the coop environment, behavioral characteristics, and egg production. This study focused on key traits of caged laying hens—including feeding, open-mouth breathing, pecking (both self and mutual), preening, and egg laying behavior. A video monitoring system was deployed to continuously track hen behavior. **Table 2** provides detailed definitions of the behavioral categories, and **Figure 2** illustrates example

behavioral images. Image annotation was performed on the filtered images using the open-source LabelImg software[24].

Table 2 Detailed description of caged laying hens

Behavioral Status	Behavioral Classification	Behavior Definition
Normal	Eat	Chicken head deep into the feeding trough
Abnormal	Open mouth	Chickens breathe with their mouths open
	Peck	Chickens peck at their own feathers
	Peckother	Chickens peck at the feathers of other chickens

a)Eat



b)Open Mouth

c)Peck

d)Peck other

Figure 2 Example of an image of the behavior of caged laying hens

2.3 Model Establishment

2.3.1 Yolov8 Network Model

Traditional object detection methods are mainly divided into shape and color based detection and machine learning classical algorithm based detection. Among them, the former may lead to a decrease in detection accuracy due to uneven color of feathers in captive laying hens, while the latter takes longer to detect chicken behavior and is not conducive to subsequent transplantation and real-time deployment. Therefore, compared to traditional detection methods based on shape and color features and detection methods based on classic machine learning algorithms, deep learning based object detection methods have better accuracy and real-time performance.

Deep learning based object detection is mainly divided into single-stage detection algorithms mainly based on YOLO(You Only Look Once) series and SSD(Single Shot MultiBox Detector), and two-stage detection algorithms mainly based on R-CNN(Region-Based Convolutional Neural Networks)[25]. The network design and computational process of the two-stage detection algorithm are relatively complex, and although the detection accuracy is high, the speed is slow; Single stage detection algorithm design is more concise and efficient, with shorter detection time but lower accuracy. In order to achieve intelligent detection and subsequent deployment of key behaviors of captive laying hens, it is required that the model can perform real-time detection and high accuracy. This paper takes the YOLOv8 model as the basic model architecture, which strikes a balance between detection accuracy and time consumption while effectively reducing computational overhead. On this basis, we carry out optimization and improvement to ensure real-time detection performance, making it well-suited for this application scenario.

YOLOv8, as a convolutional neural network-based object detection algorithm, introduces new features compared to YOLOv7 and YOLOv5. It has significant improvements in network architecture, detection speed, and accuracy, and has real-time detection capabilities, making it easy to transplant

and deploy in the future.

YOLOv8 has introduced new features and improvements based on the success of YOLOv5, further enhancing its performance and flexibility[20], mainly including:

1) In the backbone network and Neck section, the YOLOv7 ELAN design concept was referenced, and the C3 structure of YOLOv5 was replaced with a C2f structure with richer gradient flow to improve detection precision.

2) YOLOv8 replaces the detection head with a Decoupled Head structure, separates the classification head from the detection head, and also replaces Anchor Based with Anchor Free.

3) For the loss function, YOLOv8 separates the regression task from the classification task in object detection for prediction. In regression tasks, distributed focus loss (DFL Loss) and complete intersection loss (CIOU Loss) are used.

2.3.2 Improvement Strategy for Yolov8 Model

YOLOv8 is a lightweight network with outstanding comprehensive performance[26]. Firstly, the C2f Faster[27] and C2f Faster EMA (CFE) modules [28]were used to replace the C2f modules in the neck and backbone networks of the YOLOv8n model, respectively.

To increase its efficiency, we introduce a number of improvements to its architecture.The Faster Net Block constructed based on Partial Convolution (PConv) [29] can effectively reduce redundant computation and memory access. On the basis of C2f Faster, the C2f Faster EMA (CFE) module with the introduction of EMA attention mechanism can enhance spatial feature extraction in a more effective way and save computational costs.

Secondly, in the neck network of YOLOv8n, we used Dysample, an ultra lightweight and efficient dynamic upsampler that improved the performance of the model with lower computational overhead.

Finally, by combining the exponential moving average(EMA) concept with SlideLoss, the EMASlideLoss module is proposed to improve sample imbalance, enhance model detection precision, and strengthen model robustness. The network architecture of the improved YOLOv8 model is shown in **Figure 3**.

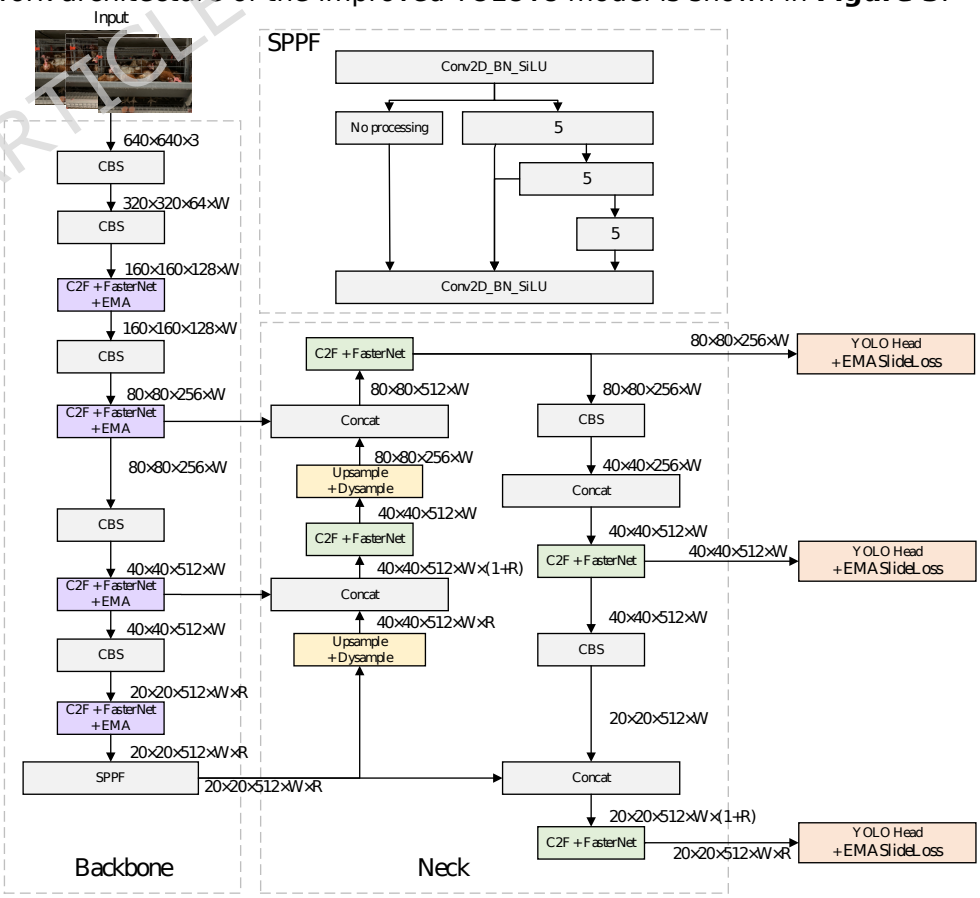


Figure 3 Improved YOLOv8 Network Architecture

In the YOLOv8 input terminal, it mainly includes input cage chicken image preprocessing, adaptive anchor box calculation, image data enhancement, etc. Among them, the cage chicken image preprocessing stage includes scaling the image size to the required size of the model, normalization, etc. Image enhancement uses the Mosaic method to randomly select 4 cage chicken images for scaling, cropping, color shaking and other operations to expand the model training dataset, thereby improving the robustness and generalization ability of the model; Adaptive anchor box calculation is achieved through learning, without the need for manual settings, to automatically calculate anchor box parameters for cage raised egg chicken images that are suitable for input. Unlike other models in the previous YOLO series, YOLOv8 turns off Mosaic enhancement in the last 100 epochs of the image enhancement stage, effectively improving the detection accuracy of the model.The image enhancement effect is shown in Figure 4.

Fig.4 Image enhancement effect

2.3.3 Fasternet Block

DWConv evolved from Conv and is currently used and plays a critical role in many mainstream neural networks. Although DWConv has high application value in reducing floating-point operations, simply replacing Conv with DWConv will have a huge impact on detection precision.In contras, PConv performs much better in floating-point operation efficiency than DWConv, occupying only a small part of conventional convolution. Therefore, the PConv module will be used in subsequent models, and its structure is shown in **Figure 5**.

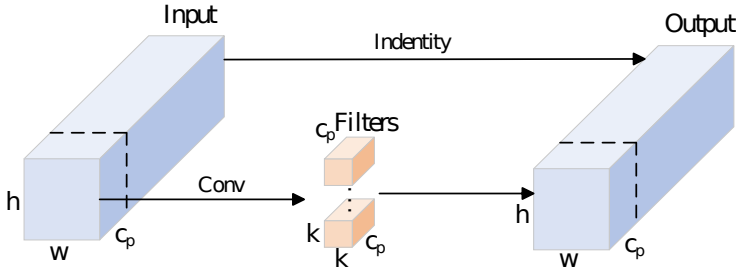


Figure 5 Schematic diagram of PConv structure

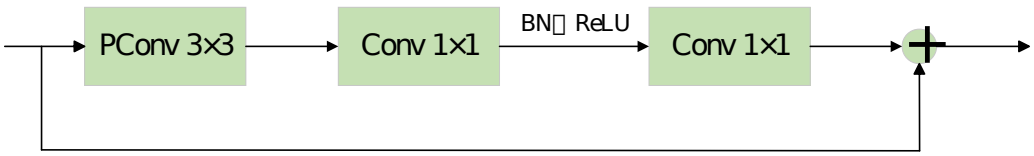


Figure 6 Schematic diagram of the FasterNet module

As shown in **Figure 6**, when the FasterNet module is used as the backbone, this model architecture will consist of four parts, each consisting of multiple FasterNet blocks, and an embedding or merging layer is set at the initial stage of each part to adjust the data flow. The last three layers in the architecture are used for feature classification processing. Each FasterNet block utilizes PConv processing internally, followed by two PWConv (Pointwise Convolution) operations. This structural design facilitates comprehensive exploration of information across all channels by the model, building a convolutional architecture similar to a "T" shape, thereby making the model more sensitive to the features of the central region. After performing the PConv operation, the normalization and activation layers are intentionally placed only after the middle layer to maintain feature richness and reduce computational latency. This study replaces the C2f layer in the neck network with the C2f FastNet module.

2.3.4 Ema - Efficient Multi Scale Attention

By introducing attention mechanisms, detection models can be guided to allocate more attention to key regions or features in images of captive laying hens, enhancing the detection performance of object detection algorithms. The EMA module does not reduce the channel dimension, but processes the channel dimension by reconstructing some channels and splitting them into several subsets to ensure a balanced distribution of spatial semantics among each subset. In addition, this module not only fine tunes the channel weights in each parallel sub-branch by encoding global information, but also enhances the output of the two parallel sub branches through cross-channel feature fusion to capture fine-grained pixel level relationships. The parallel sub branch network structure adopted by the EMA module avoids the phenomenon of excessive depth and more sequential processing. It aims to optimize the protection mechanism of global information,and preserve information while reducing computational pressure. The structure of the EMA module is shown in **Figure 7**.

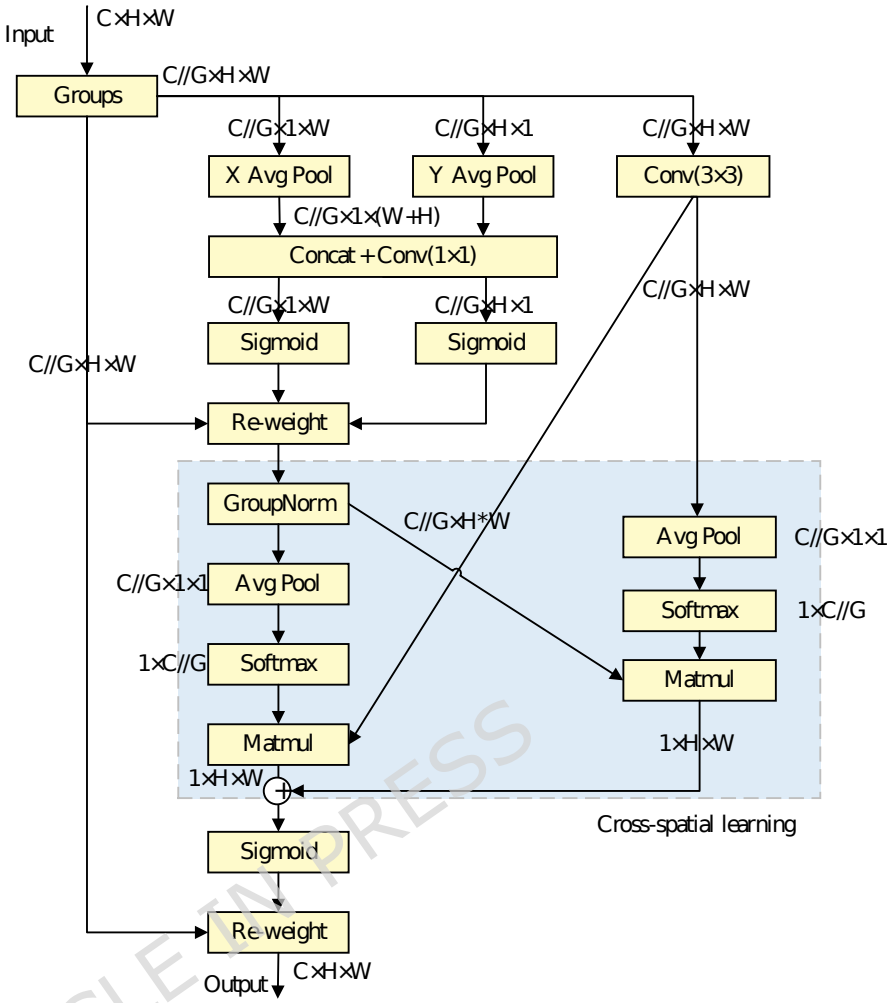


Figure 7 Schematic diagram of the attention mechanism of EMA

The EMA module achieves effective feature enhancement by using attention mechanisms to dynamically learn the importance of different regions in images of captive laying hens.It includes two main steps: generating feature maps and computing attention mechanisms. In the initial stage, convolutional layers are used to transform the input images of captive laying hens into many different scales, thereby obtaining a series of feature maps at different scales. Subsequently, by averaging and maximizing the pooling of feature maps at various scales, pooled feature representations are constructed, and based on these pooled features, attention scores for each scale feature are further calculated, which represent the importance of the features at each scale. Finally, the original feature map and its corresponding attention scores were combined and subjected to weighted synthesis analysis, further obtaining a comprehensive multi-scale feature representation.

The EMA module can not only automatically learn the features of various scales in the input images of captive laying hens and calculate attention weights, but also perform adaptive fusion. In addition, EMA also has the advantage of cross spatial learning, which can not only improve model detection efficiency but also obtain feature representations at different scales. By utilizing the flexibility and lightweight features of EMA, it was integrated into the FasterNet Block to form the C2f FasterNet EMA module. This study replaces the C2f module in the backbone network with the C2f FasterNet EMA module to enhance the feature fusion capability of the backbone network.

2.3.5 Neck Improvement

The Dysample upsampler completes the upsampling task from the perspective of learning sampling, effectively avoiding the high time cost of

dynamic convolution operations and additional self networks. Compared to traditional kernel based dynamic samplers, the Dysample upsampler does not require specialized CUDA support and has advantages such as fewer parameters, fewer floating-point operations, less GPU memory usage, and lower latency. Based on this, the Dysample upsampler has demonstrated a higher advantage in dense prediction such as object detection and semantic segmentation.

The sampling set S of Dysample is composed of the original sampling grid and its generated offsets. The formula for upsampling Dysample is:

$$S = g + o(2-1)$$

In the formula, g is the original sampler and o is the offset. The generation of offset is determined by a linear+pixel beam, and its range is determined by two methods: static factor and dynamic factor. In the static factor method, the author uses 0.25 times the offset to represent the theoretical boundaries of non overlapping and overlapping, in order to avoid differences in boundary predictions caused by overlapping local S-shaped positions. The calculation formula for the static range factor is shown in **Equation (2-2)**. In the dynamic factor method, the dynamic range factor is formed by utilizing the features of linear projection input to improve the flexibility of offset. The calculation formula is shown in **Equation (2-3)**.

$$o = 0.25\ \mathrm{linear}(\chi)(2-2)$$

$$o = 0.5\mathrm{sigmoid}(\mathrm{linear}_1(\chi))\cdot\mathrm{linear}_2(\chi)(2-3)$$

Taking the static factor sampling method as an example, given that the size of the feature map is C × W × H and the upsampling factor is s, the feature map is input to a linear layer with input channel C and output channel s2. The size of the feature map is also converted to 2 × sH × sW through pixel flow, where 2 represents the coordinates of x and y. Finally, using **Equation (2-4)**, the feature map size becomes C × sH × sW. Thus, an upsampling feature map with a size of C × sH × sW is generated.

$$\chi' = \mathrm{grid}_{\mathrm{sample}(\chi,S)}(2-4)$$

The working principle of the Dysample upsampler is shown in **Figure 8**.

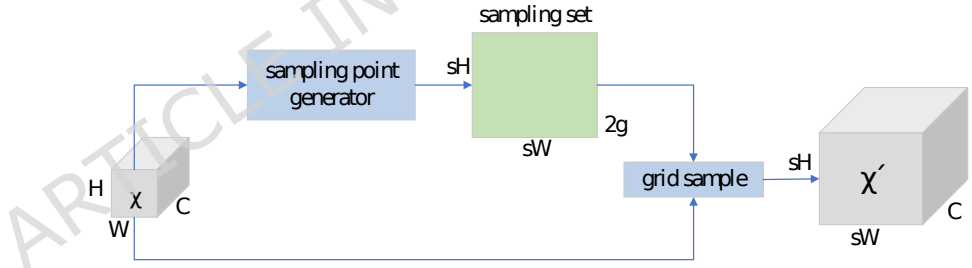


Figure 8 Schematic diagram of how Dysample upsampling works

During the sampling process using the Dysample upsampler, the input images or feature maps of captive laying hens are first processed through a series of previously designed upsampling modules, which can adaptively and dynamically adjust the sampling mechanism based on the input images. According to the different characteristics and requirements of the input data, the sampler adopts different sampling strategies to improve the efficiency of upsampling and achieve better sampling results.

2.3.6 Improvement of Loss Function

The loss function evaluates the detection ability of a detection model on a training dataset by measuring the difference between the true label and the predicted label[26]. After calculating the loss value, the model fine tunes the model parameters using the backpropagation algorithm to fit the training data and ultimately achieve the goal of obtaining a highly accurate training model.

1) CIOU+DFL loss

Yolov8 uses a combination of CIOU loss and DFL loss to calculate the regression loss function of the model. For bounding box regression and matching, we adopt the Complete Intersection over Union (CIoU) loss[30], which extends IoU-based loss by incorporating the normalized distance between bounding box centers and the aspect ratio consistency constraint to address the slow convergence issue of traditional IoU loss in non-overlapping

box scenarios. Due to the introduction of Anchor Free based on the center point in the YOLOv8 object detection model, the output parameters of the model have been changed from off set (anchor box size offset) to “ltrb=left, top, right, bottom” (predicting the distance between the left, upper, right, and lower borders of the target box and the center point of the target box). Therefore, in order to adapt to the application of Anchor Free and improve the generalization ability of the key behavior detection model for captive laying hens, DFL loss is introduced. DFL loss is achieved through cross entropy, allowing the network model to quickly concentrate on the target box and surrounding adjacent regions.

The calculation formula for CIOU loss is as follows:

$$L_{CIOU} = 1-IOU + \frac{\rho^2(b,b^{gt})}{c^2} + (2-5)$$

In the formula $\rho^2(b,b^{gt})$ represents the Euclidean distance between the center of the predicted box and the center of the true box, c represents the diagonal distance that can simultaneously cover the minimum rectangular box area donated by both the predicted box and the true box, v is the aspect ratio of the predicted box, the formula to calculate is as follows:

$$\alpha = \frac{v}{1-IOU + v}(2-6)$$

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2(2-7)$$

The calculation formula for DFL loss is:

$$DFL(S_i,S_{i+1}) = -((y_{i+1}-y)\log(S_i) + (y-y_i)\log(S_{i+1}))(2-8)$$

In the formula, $y_i \in \{y_0,y_1,\ldots,y_{n-1},y_n\}$, represented as uniform sampling within the possible intervals of y $[y_0,y_n]$.

2) BCE loss

YOLOv8 continues to use the binary cross entropy BCE in YOLOv5 to calculate its classification loss, making separate judgments for each category and outputting the confidence level of the response. Finally, the maximum value obtained is used as the confidence level of the anchor box. The calculation formula is as follows:

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -[y_i \cdot \log(p_i) + (1-y_i)\cdot\log(1-p_i)](2-9)$$

In the formula, p_i is the model detection result label, and y_i is the real label.

3) EMASlideLoss loss

The problem of sample imbalance often occurs in model training, where in large datasets, the number of simple samples far exceeds that of difficult samples. This phenomenon has attracted widespread attention. Most traditional classification loss functions only predict the category of objects, without considering their specific location and scale. In practical applications of object detection, accurately identifying object categories located at different positions and scales is crucial. In order to compensate for the deficiency of traditional classification loss, the author of YOLOFace V2 proposed the SideLoss loss function, which adopts a sliding window approach to divide the input image into several small regions and calculate the classification loss separately for each small region. This method enables the detection model to better obtain the classification of objects at different positions and scales in the images of captive laying hens, thereby improving the prediction performance and detection precision of the model. Compared with traditional classification loss, SideLoss loss function introduces object position and scale information as consideration factors in addition to category information during object detection, making it more effective in addressing the challenges of scene complexity and scale diversity in detecting targets.

The Intersection over Union (IOU) between the real box and the predicted box is the basis for determining the difference between simple and difficult samples. Specifically, in SideLoss, in order to simplify the number of hyperparameters, the intersection to union ratio of all bounding boxes is calculated and averaged, and represented as the threshold μ. Samples below or above this threshold are considered negative or positive. In model training, samples around the boundaries are often severely affected due to unclear classification. At the same time, these sample sizes are relatively small. If we want the model to fully learn from these types of samples and use them for better training, we need to increase the weights assigned to these samples.

Highlighting the importance of using a Slide weighting function on the samples at the boundary after dividing them by a threshold as mentioned earlier. The slide weighting function is shown in the following equation:

$$f(x) = \begin{cases} 1 & x \le \mu - 0.1 \\ e^{1-\mu} & \mu < x < \mu - 0.1 \\ e^{1-x} & x \ge u \end{cases} \quad (2-10)$$

Where f(x) is a sliding function operation, x is the IoU between the predicted box and the truth during training, and mu is the weight threshold.

The EMASlideLoss loss function combines the strategy of EMA (Exponential Moving Average) with SideLoss. During the training phase of the detection model, the EMASlideLoss loss performs an exponential moving average operation on the loss weights calculated by the SideLoss loss function to smooth the loss curve. This processing method is beneficial for reducing the loss fluctuations and random noise generated during training, in order to enhance the stability of key behavior model training for captive laying hens and optimize the model's generalization ability. The EMA calculation formula is as follows:

$$v_t = \beta \cdot v_{t-1} + (1-\beta) \cdot \theta_t \quad (2-11)$$

In the above equation, θ_t represents the model weight values (weights) obtained from the t-th update, and v_t represents the moving average of all parameters from the t-th update, that is, the shadow weights; β is a weighted weight value, usually set to 0.9~0.999.

2.4 Evaluation Metrics

In this paper, the performance of the key behavior detection model for caged laying hens is evaluated by calculating precision (P), recall (R), accuracy (A), F1 - score, mean average precision (MAP), and detection speed. Among them, precision is used to evaluate the accuracy of the model in detecting the key behaviors of caged laying hens; recall evaluates the model's ability to accurately detect the key behaviors of caged laying hens; accuracy is the proportion of correct detections of the key behaviors of caged laying hens by the model in the samples; mean average precision is the mean value obtained by calculating AP; detection speed evaluates the time taken by the model to detect the key behaviors of caged laying hens in a single image; the F1 - score is a comprehensive calculation of recall (R) and precision (P). These parameters are used to comprehensively evaluate the model's ability to detect the key behaviors of caged laying hens. The formulas for obtaining the above - related parameters are as follows:

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (2-12)$$

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (2-13)$$

$$F1 = \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (2-14)$$

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (2-15)$$

$$AP = \sum_{k=0}^{k=n-1} [\text{Recalls}(k) - \text{Recalls}(k+1)] * \text{Precision}(k) \quad (2-16)$$

$$MAP = \frac{1}{n} \quad (2-17)$$

In the above formulas, TP represents true positives, that is both the true label and the predicted label are 1; TN represents true negatives, that is both the true label and the predicted label are 0; FN represents false negatives, that is the true label is 1 and the predicted label is 0; FP represents false positives, that is the true label is 0 and the predicted label is 1. n is the number of categories of key behaviors of caged laying hens. AP_k is the value when the category is k.

3 Results

Using the cage raised egg chicken dataset created in the previous section, five models with different depths in the YOLOv8 series were compared and trained to ultimately obtain the detection model with the best detection performance. The establishment of a training environment is the foundation of model training. Therefore, in the preparation work for object detection model training, it is necessary to build the model framework and configure the training environment on the training equipment. The specific parameter

settings are shown in **Table 3**.

Table 3 Parameters for model training

Configuration Type	Name	Configuration Parameters	Details
Hardware	Deep Learning Machine	NVIDIA RTX4090	Video Memory 24G
	CPU	13th GenIntel@ Core" i9-13900K Fx 32	—
Software	Operating System	Ubuntu 22.04.3 LTS	x64
	Python	3.8	—
	CUDA	11.7	—

Due to the comparison of model detection performance in the future, in order to avoid the impact of inconsistent parameters on the training effect of the model, the hyperparameters set in the model framework should be consistent. The hyperparameter settings for model training are shown in Table 4 below：

Table 4 Parameter settings for model training

Parameter Name	Parameter Type	Parameter Value	Remarks
lro	float	0.01	Initial Learning Rate
lrf	float	0.01	Final Learning Rate
amp	bool	True	Mixed Precision Training
conf	float	0.25	Target Detection Confidence Training
warmup_epochs	int	3	Preheating Training Frequency
warmup_momentum	float	0.8	Preheating Training Initial Quantity
imgsz	int	640	Enter Image Size
batch_size	int	16	Enter Batch Size

3.1 Experimental Comparison with Other Models

The comparison of model sizes and parameters for five different versions of YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x in the YOLOV8 series is shown in **Table 5**:

Table 5 Comparison results of the basic models

Model	Gflops/G	Params /M	Number of Layers
YOLOv8n	8.9	3.16	225
YOLOv8s	28.8	11.17	225
YOLOv8m	79.3	25.90	295
YOLOv8l	165.7	43.69	365
YOLOv8x	258.5	68.23	365

According to the analysis in the table above, it can be seen that YOLOv8n has the smallest number of parameters, network layers, and the Gflops are 3.16M, 225 layers, and 8.9G, respectively. In comparison, the Gflops of YOLOv8s are 28.8G, which is three times that of YOLOv8n, and its parameter count is approximately four times that of YOLOv8n. The Gflops of YOLOv8x are nearly 30 times that of YOLOv8n, and the parameter count is 22 times that of YOLOv8n. Considering that the model needs to be easy to transplant and deploy in the future, and there is a high demand for lightweight models, YOLOv8n is selected as the basic model for key behavior detection of captive laying hens.

In order to further demonstrate the advantages of the YOLOv8 model, a comparative experiment was conducted between YOLOv8 and some mainstream object detection models while maintaining consistency in the training environment, dataset, and training parameters. Select evaluation indicators from precision, recall mAP50, Gflops the comparative experimental results in terms of params, network layers, and FPS are shown in **Table 6&Figure 9**.

Table 6 Comparison of training results of different models

Model	Precision P/%	Recall R/%	Mean Average Precision mAP/%	Gflops	Params /M
YOLOv5n	86.88	88.16	87.83	2.32	6.5
Fast R-CNN	56.26	89.22	34.67	140.53	354.8
SSD	51.78	89.07	30.85	168.26	386.7
YOLOv8n	88.42	87.8	89.52	3.01	8.1

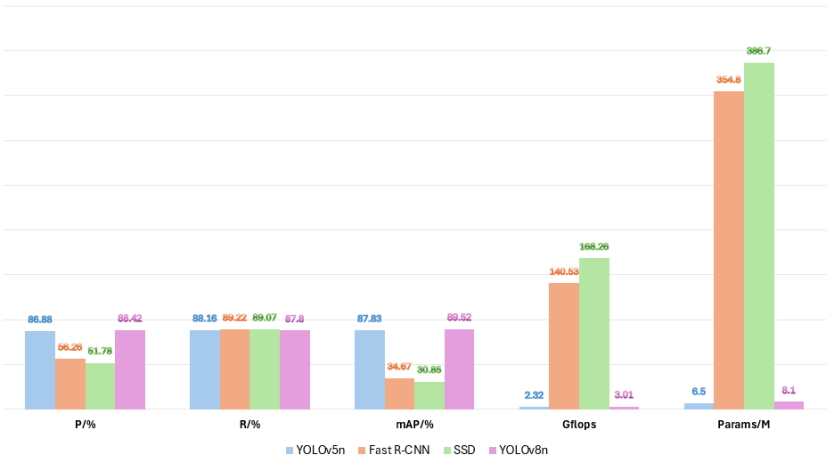


Figure 9 Comparison of training results of different models

As shown in the **Table 6**, the average accuracy of Fast R-CNN and SSD detection in detecting key behaviors of captive laying hens is relatively low, at 56.26% and 51.78%, respectively, while the parameter quantities are as high as 354.8M and 386.7M, respectively. As shown in **Figure 10**, the average accuracy of the four models detected from lowest to highest is 30.85%, 34.67%, 89.83%, and 89.52%, respectively. Among them, YOLOv8n has the highest accuracy, reaching 89.52%, and YOLOv5n has the smallest parameter size of 6.5G. Although the Gflops of YOLOv8n is 3.01G, slightly higher than that of YOLOv5n (2.32G), the small difference in parameter size has little effect on the detection performance of the model. Therefore, in the future, YOLOv8n will continue to be used for the detection and recognition of key behaviors in captive laying hens, and further optimization and improvement will be made.

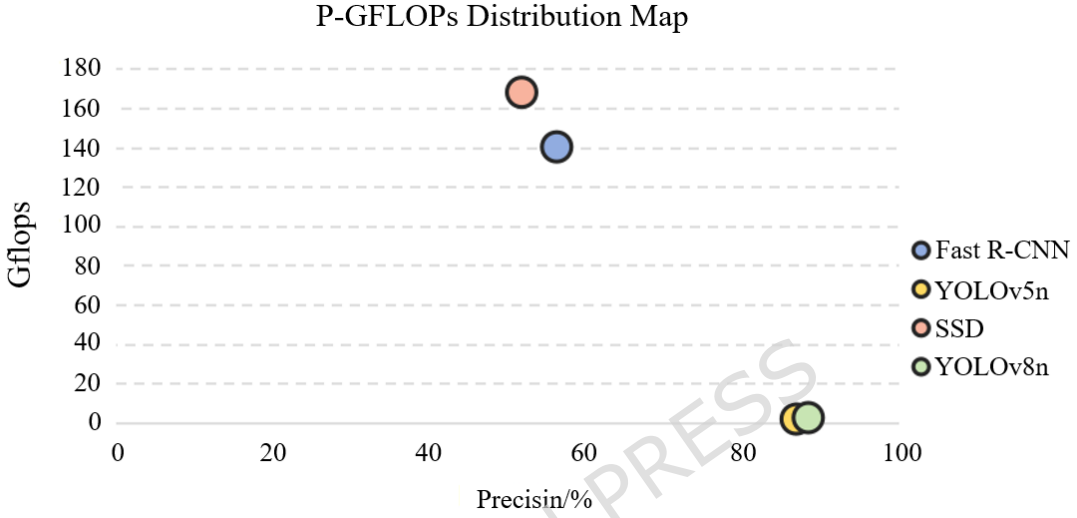


Figure 10 Distribution of P-GFlops in the model

To comprehensively improve the detection performance of YOLOv8, this section will improve and optimize the original model. The improvement schemes are as follows: (A) C2f FasterNet EMA module replaces the C2f module in the backbone to enhance the model's feature fusion ability and improve detection efficiency. (B) The C2f FasterNet module replaces the C2f module in the neck, reducing computational latency while maintaining feature richness. (C) Improve the upsampling mechanism in the neck by using a Dysample upsampler. (D) Simultaneously incorporating the loss function EMASlideLoss. Integrate each improvement scheme into the original model for training, construct 5 sets of experiments to verify the optimization effect of the module. The comparison of the performance of several improved models and the original model in detecting key behaviors of captive laying hens is shown in **Table 7**.

Table 7 Comparison results of ablation experiments with different models

A	B	C	D	mAP50 /%	mAP50~95 /%	Params /M	GFlops/G
×	×	×	×	89.52	72.03	3.01	8.1
√	×	×	×	91.31	73.2	2.65	7.1
√	√	×	×	90.59	74.51	2.31	6.4
√	√	√	×	91.49	72.41	2.32	6.4
√	√	√	√	91.78	78.26	2.32	6.4

Note: "√" represents the model using this module; The 'x' indicates that the model does not use this module.

As shown in the table above, the original YOLOv8n model has an mAP50 of 89.51%, mAP50~95 of 72.03%, parameter size of 3.01 × 106 M, and GFlops

of 8.1G for detecting key behaviors in captive laying hens. After introducing the FasterNet module and EMA attention mechanism into the backbone, the network's feature extraction ability was improved, with an average detection accuracy increase of 1.79 percentage points and a parameter increase of 23.48M. And when the FasterNet module is added to the neck layer, the complexity of the model is reduced. Although mAP50 has a lower attention mechanism compared to adding EMA, it is 1.07% higher than the original model. On the basis of the first two, the sampler in the neck layer was improved by introducing the Dysample upsampler, resulting in an mAP50 of 91.49% and a parameter count of 23.18M. At the end, the EMASlideLoss loss function was added to accelerate the convergence speed of the model. The results showed that the improved models mAP50 and mAP50~95 achieved the highest values compared to the first four groups, with 91.78% and 78.26%, respectively, which were 2.26% and 6.23% higher than the original model, 20.17M higher than the original model, and 1.7G lower in GFlops.

In summary, the improved YOLOv8n model has higher sensitivity, better detection performance, and more stable detection of key behaviors in captive laying hens. Therefore, the improvement plan proposed in this study has high feasibility.

3.2 Advantages of the Improved Model

3.2.1 Core Functions of the Improved Modules

The Backbone layer incorporates the C2f-FasterNet-EMA module, where the original C2f module is replaced by FasterNet's lightweight convolutional structure. This reduces parameter count while retaining more high-frequency details such as feather textures and behavioral postures. The EMA (Efficient Multiscale Attention) mechanism enhances focus on critical regions like the beak through cross-channel weighting, effectively suppressing interference from complex backgrounds.

In the Neck layer, the C2f-FasterNet module optimizes computational latency via FasterNet's dynamic sparse convolution strategy. By activating convolutional kernels only in active regions of densely stacked cage-raised chicken scenarios, inference speed improves by approximately 15% while preserving feature map channel dimensions. Residual connections maintain multi-scale features, preventing loss of fine-grained information caused by increased network depth—crucial for distinguishing subtle motion differences between "self-pecking" and "mutual pecking" behaviors.

Compared to traditional upsampling methods prone to edge blurring, the Dysample upsampler generates adaptive upsampling weights through dynamic kernel prediction. Experimental results demonstrate a 21.3% reduction in bounding box localization error for small targets like "open-mouth"

The EMASlideLoss function dynamically adjusts loss weights for challenging samples (e.g., occluded chickens) using Exponential Moving Average (EMA), preventing overfitting to simpler samples. A sliding window mechanism balances positive-negative sample ratios, addressing missed detections in densely packed cage environments.

3.2.2 key Principles of Accuracy Improvement

The EMA attention mechanism in the Backbone layer and the multi-scale feature fusion in the Neck layer synergize to achieve multi-scale feature collaboration. Specifically, the EMA mechanism strengthens focus on critical regions (e.g., chicken beaks) through channel-wise attention, while C2f-FasterNet preserves spatial details via residual connections.Their integration significantly enhances confidence in behavior classification.

FasterNet's depthwise separable convolution reduces parameters by 30%, yet retains effective features through dynamic kernel computation. For instance, when detecting “feeding” behavior, the model achieves a 19.8% improvement in inter-channel feature correlation while reducing parameters by 7%, balancing efficiency and accuracy.

The dual dynamic optimization of Dysample and EMASlideLoss ensures robustness in complex scenarios: Dysample adaptively adjusts upsampling kernels based on local texture complexity, while EMASlideLoss modifies

gradient updates according to sample difficulty. This dual mechanism markedly improves performance under challenging conditions like varying lighting and overlapping chickens in stacked cage environments.

As shown in **Figure 11**, the improved feature enhancement module greatly enhances the detection of small targets such as chicken beaks. Computational optimization reduces inference delay in dense populations without sacrificing feature richness, and the dynamic adjustment mechanism mitigates false positives/false negatives in occluded scenes. These three aspects work together to improve the average detection accuracy of key behaviors, verifying feature enhancement, computational efficiency, and dynamic robustness.

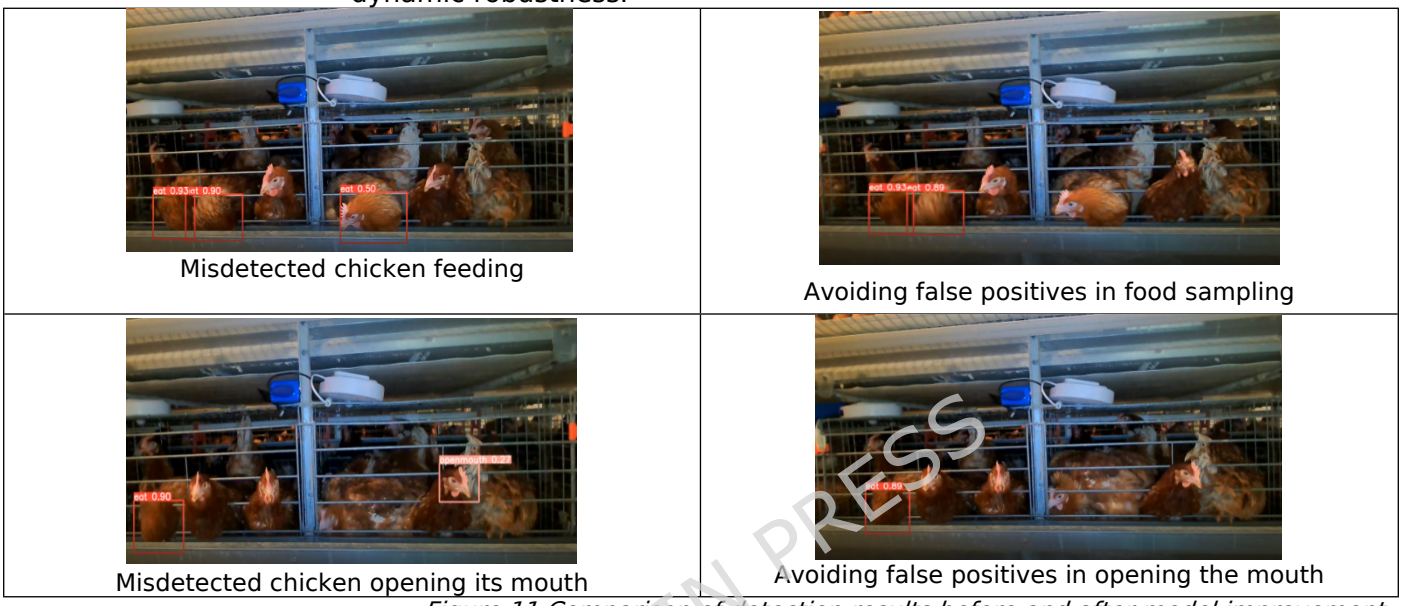


Figure 11 Comparison of detection results before and after model improvement

3.3 Model Performance

The confusion matrix provides an intuitive visualization of the model's performance across each behavior category. In this study, a normalized confusion matrix is utilized to evaluate the model's effectiveness. The confusion matrix of the improved YOLOv8n model for key behavior detection in cage-raised laying hens is illustrated in **Figure 12**.

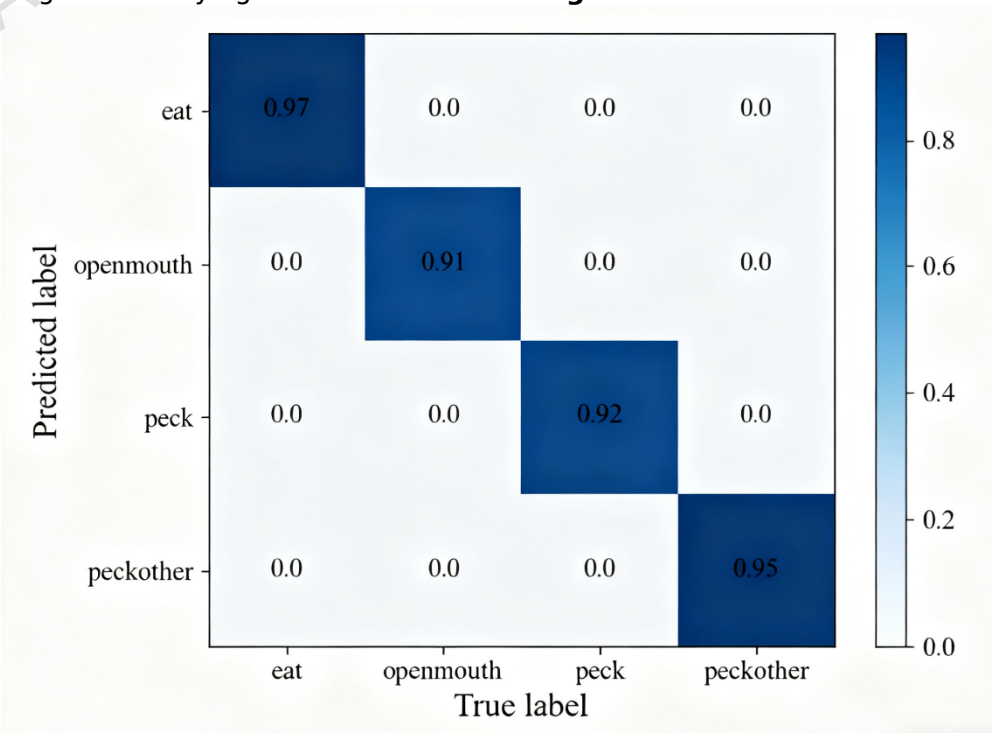


Figure 12 Improved confusion matrix of the YOLOv8n model

In this study, a normalized confusion matrix was used to evaluate the effectiveness of the model. The confusion matrix of the improved YOLOv8n model used for key behavior detection in captive laying hens is shown in Figure 11. The matrix shows that the model exhibits excellent performance in four behavior categories, eat, open mouth, peck, and peckether, with recall rates of up to 0.97, 0.91, 0.92, and 0.95 for each category, respectively. And all non diagonal elements have zero values, indicating that the model did not make any misjudgments in recognizing chicken behavior in this experiment and achieved inter class discrimination. Based on this, a brief analysis of the results shows that the eat behavior pattern is relatively unique, usually characterized by the head staying near the food trough for a long time, accompanied by continuous pecking movements, with significant differences in duration and posture compared to brief "pecking" or "opening the mouth". In addition, pecking behavior also has more obvious characteristics compared to other behaviors, providing a guarantee for the strong discriminative power of the model.

The detection results of the improved model are presented in **Table 8.** According to the performance metrics, the mean Average Precision (mAP50) for the behaviors of eating, pecking, and pecking othersall exceed 93%, and their Recall (R) rates are all above 92%, indicating that the model can effectively capture the visual features of these behaviors.

However, the recognition performance for the open-mouthbehavior shows a noticeable gap compared to other categories, with an mAP50 of 81.03% and a Precision (P) of 73.7%, which are significantly lower.

This discrepancy may originate from the following reasons: firstly, the transient nature of the open-mouthaction bears high visual similarity to the beak movement observed during peckingbehavior. Consequently, the model may struggle to learn highly discriminative features from single frames or short temporal sequences, leading to misclassification.

Secondly, challenges such as frequent occlusions among birds in enriched cage environments further complicate the extraction of subtle motion features like beak opening.

Although the recall rate for open-mouthreaches 91.3%, suggesting a satisfactory detection rate for true instances of this behavior, the current experimental evaluation possesses certain limitations.

The presented results are primarily point estimates (e.g., mAP50, P, R), lacking quantification of the reliability of performance differences through statistical significance tests or reporting the standard deviation of metrics via repeated cross-validation.

This makes it difficult to rigorously conclude whether the improved recognition performance across all behaviors—especially whether the lower mAP50 for open-mouthis significantly weaker than that of other behaviors—holds statistical significance rather than stemming from random variations in the training data.

For instance, hypothesis testingis required to provide statistical evidence.To enhance model robustness, particularly its discriminative capability for easily confusable behaviors, future work could consider improvements in the following aspects:

(1) Introduce temporal modeling capabilities by employing network architectures such as 3D CNNs or Transformers to analyze behavioral sequences rather than relying solely on static frame features, thereby capturing the differences in action coherence between open-mouthand pecking.

(2) Apply model optimization strategies like knowledge distillation or attention mechanisms (e.g., the SEAM module) to enhance the model's focus on critical subtle features and suppress interference from complex backgrounds.

(3) At the data level, implement data augmentation specifically for challenging samples like open-mouth, and ensure that the evaluation process includes repeated experiments or cross-validation to provide stability measures (e.g., standard deviation) and conduct statistical tests, thereby enabling more rigorous inferences regarding model performance.

Table 8 Improved YOLOv8n training results

Type	P/%	R/%	mAP50/%	mAP50-95/%
eat	96.27	94	98.15	85.83
openmouth	73.7	91.3	81.03	70.75
peck	99.92	92.31	93.65	78.06
Peck_other	91.2	95.45	94.32	78.4

Figure 13 Visualizing the heat map of model training

The improved YOLOv8n detection heatmap for four behaviors, including eating, open mouth breathing, pecking, and peckother, is shown in **Figure 13.** As the IOU increases, the average accuracy of each type of behavior gradually decreases. The average accuracy of the feeding behavior did not change much before the IOU was 0.80, and the decline rate was relatively slow, remaining above 95%. The average accuracy of mouth opening behavior remains unchanged at 81.03% before IOU=0.7, which is lower than the other three behaviors and has poor detection performance; When the IOU is within the two ranges of 0.50~0.70 and 0.75~0.90, there is no change in the average accuracy of self pecking behavior with increasing IOU. The average accuracy of mutual pecking behavior begins to decrease after IOU=0.80, and the rate of decrease is relatively fast, ultimately decreasing to 5.35% when IOU is 0.95. At IOU=0.95, the average accuracy of feeding, mouth breathing, self pecking, and mutual pecking behaviors were 17.45%, 13.84%, 21.54%, and 5.35%, respectively. The detection accuracy was low and the detection effect was unsatisfactory. Therefore, under strict detection conditions, it may be difficult to achieve accurate detection of the four behaviors. An example of its actual detection effect is shown in **Figure 14**.

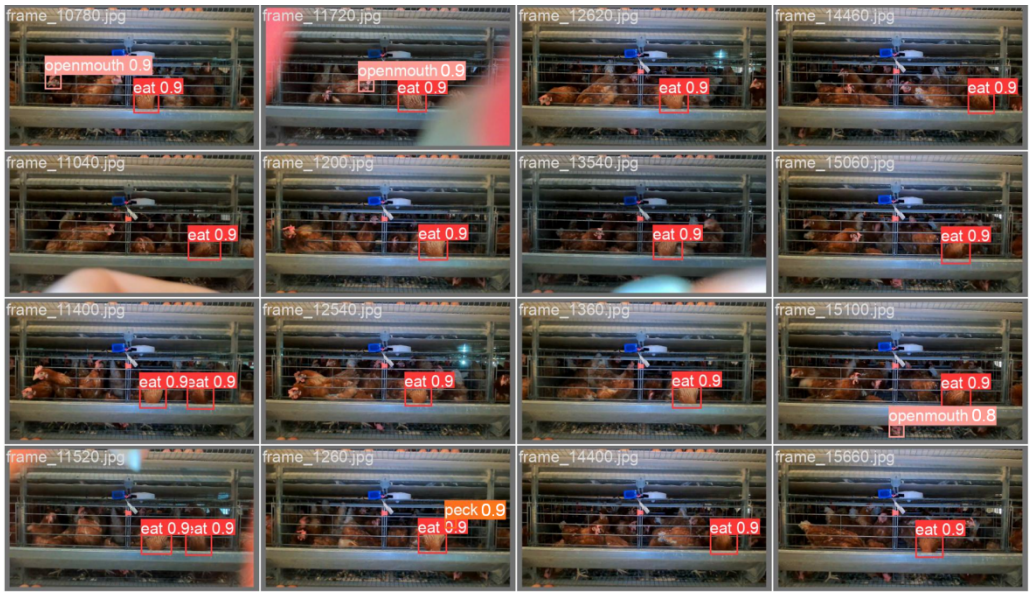


Figure 14 Example of detection effect

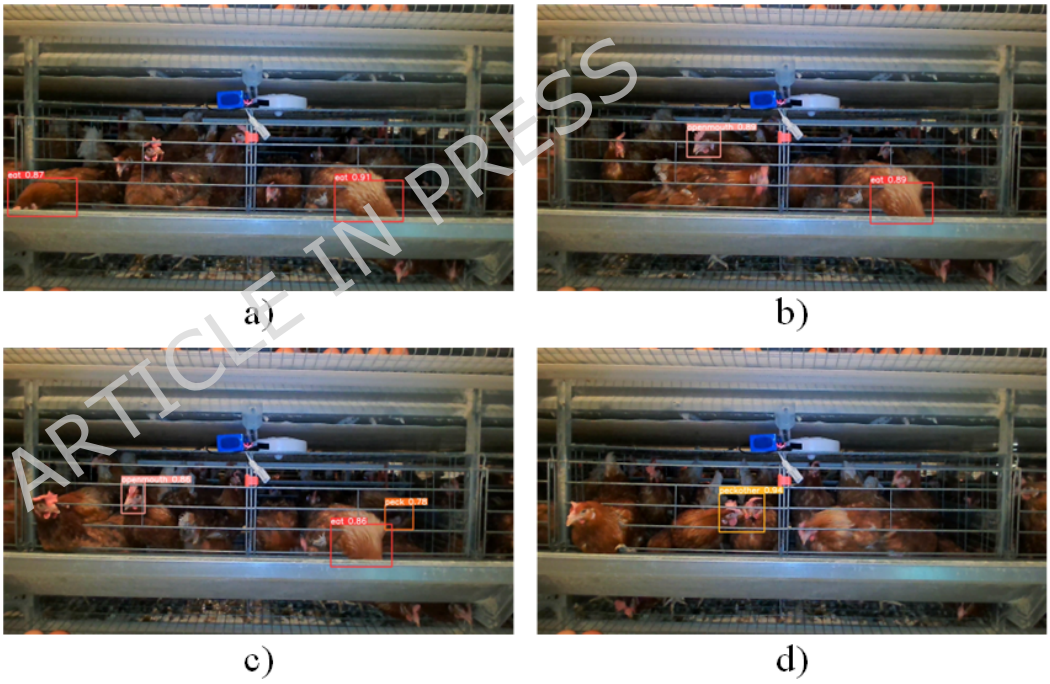


Figure 15 Different types of detection results

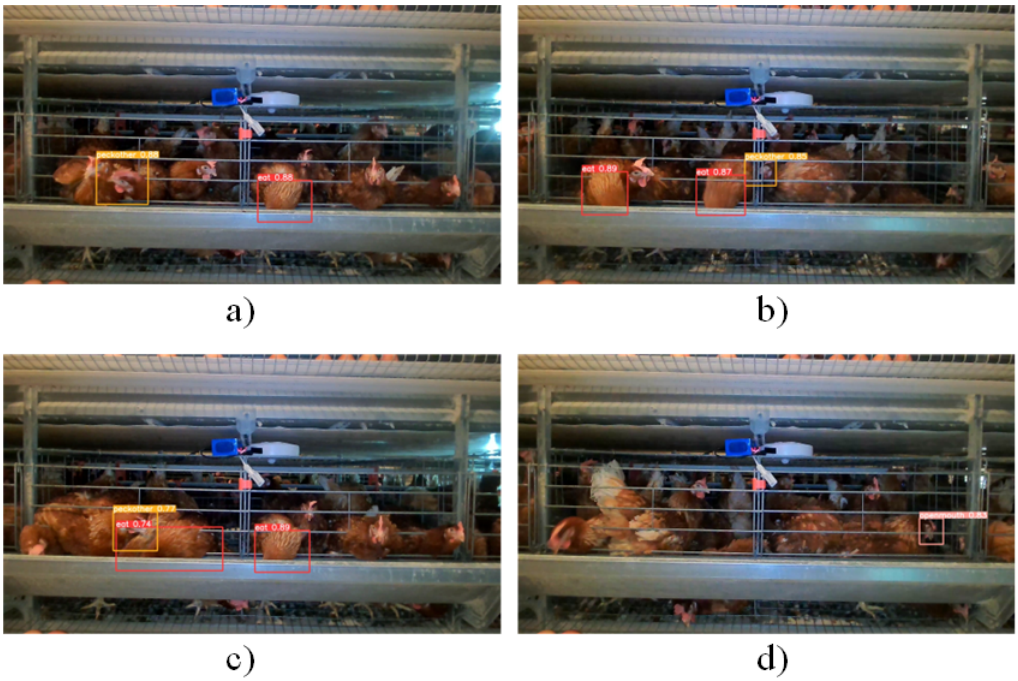


Figure 16 Detection results in complex scenes

Figure 15 shows the detection results under different types, where a) is eat; b) Open mouth when the cage is obstructed; c) For the case of severe occlusion, d) is the peck other. **Figure 16** shows the recognition results in a more complex scene, where a) is a Peck with a large number of chicken feathers as the background; b) Peck severely obstructed by cages; c) Peck other for complex chicken activity backgrounds; d) Open mouth for poor visual angles. According to the detection results, this model can recognize chicken behavior in obstructed and complex scenes well and classify them correctly, effectively avoiding missed or false detections, and has certain application value.

In summary, the improved YOLOv8n performs well in detecting feeding, self pecking, and mutual pecking behaviors. However, the detection of mouth opening breathing has slightly lower accuracy due to the small size of the target and severe occlusion.

4 Discussion

This study proposes an enhanced YOLOv8-based detection framework tailored for practical behavior monitoring in caged-layer production systems. The results demonstrate that the integration of C2f-FasterNet-EMA modules and the Dysample upsampler can effectively address common engineering challenges in poultry houses, such as limited computing resources, complex visual backgrounds, and the need for real-time processing. The improved model shows substantial advantages in detecting subtle and health-related behaviors, particularly self-pecking and mutual pecking, which are important indicators of welfare and flock stability. The reduction in model size further suggests strong feasibility for deployment on edge devices or mobile inspection robots frequently used in smart-farming environments.

Despite these strengths, the system remains limited in distinguishing visually similar short-duration behaviors (e.g., open-mouth breathing). This core challenge stems primarily from three key factors: frequent occlusions in real-world capture scenarios, motion overlap between adjacent behavioral actions, and the inherent lack of temporal contextual information in the single-frame detection framework. Furthermore, the study's dataset is derived from a single farm environment and a single poultry breed, which restricts the model's generalizability to diverse rearing systems, variable lighting conditions, and breed-specific behavioral patterns. Collectively, these constraints underscore the critical need for expanded data collection across multiple

geographically distinct farms, varied housing structures, and different breeds—along with diverse environmental perturbations—to substantially enhance the model's cross-domain robustness and real-world applicability.

Though this work achieves strong performance for our target livestock behavioral recognition task, it has notable limitations that highlight core technical and practical challenges in designing and deploying such systems in real-world settings; these constraints, first referenced in the Results section, are expanded upon here for critical discussion. A primary limitation is the lack of explicit modeling of spatiotemporal dependences between sequential action frames: the current single-frame detection framework focuses solely on spatial feature extraction and does not capture dynamic temporal correlations or spatial contextual interactions across consecutive video clips. This deficiency directly exacerbates a core practical challenge: the system remains limited in distinguishing visually similar short-duration behaviors (e.g., open-mouth breathing). This latter issue stems from three intertwined factors—frequent occlusions in real-world farm capture scenarios, motion overlap between adjacent behavioral actions, and the inherent absence of temporal contextual information in single-frame inference—all of which are compounded by the lack of spatiotemporal feature fusion in the current framework. Furthermore, the study's dataset is derived exclusively from a single farm environment and a single poultry breed, which further restricts the model's generalizability to diverse rearing systems, variable on-farm lighting conditions, and breed-specific behavioral patterns that deviate from the study's experimental setting. Collectively, these constraints underscore two critical imperatives for future work: first, the need to integrate lightweight spatiotemporal fusion modules into the framework to capture sequential contextual information, thereby mitigating the challenges of distinguishing visually similar short-duration behaviors without compromising real-time inference efficiency; second, the urgent requirement for expanded data collection across multiple geographically distinct farms, varied housing structures, different poultry breed, and diverse environmental perturbations. Such efforts will not only address the core technical challenge of spatiotemporal dependence modeling but also substantially enhance the model's cross-domain robustness and real-world applicability across unconstrained farm settings.

Future work will focus on three interconnected directions to enhance the system's performance, practicality, and scalability—while strictly preserving its lightweight, real-time deployment capability for intensive poultry production. First, to capture dynamic spatiotemporal cues critical for distinguishing visually similar short-duration behaviors, we will integrate efficient temporal modeling strategies (rather than computationally heavy 3D CNNs or standard transformer-based networks). Specifically, we will leverage knowledge distillation frameworks [31] (validated for lightweight animal action recognition) and selective state architectures (e.g., Mamba) [32]—approaches that enable effective capture of sequential dependencies with minimal computational overhead, aligning with our core focus on model efficiency. Second, controlled disease-induction experiments will be conducted to establish quantitative correlations between subtle behavioral deviations and early-stage health deterioration. This will elevate the system from a passive behavioral detection tool to an active, proactive health prediction platform, addressing a key unmet need in precision poultry farming. Third, multimodal sensing integration—including thermal imaging (for physiological state inference) and acoustic data (for non-visual behavioral cues)—will be explored to enhance recognition reliability under challenging field conditions (e.g., low lighting, occlusion, or high stocking density). Collectively, these extensions build on the proposed framework's strengths, reinforcing it as a promising, scalable solution for automated, real-time behavioral monitoring in intensive poultry production systems.

5 Conclusion

Ablation experiments by sequentially incorporating the improved strategies into the original YOLOv8n model demonstrate that the combination of all three enhancements yields the best detection performance. The detection accuracies for feeding, open-mouth breathing, self-pecking, and

mutual pecking reach 98.15%, 81.03%, 93.65%, and 94.32%, respectively. The mAP@0.5 improves by 2.26% compared to the baseline model, while the model size decreases by 22.92%, simultaneously meeting practical requirements for real-time detection and accuracy in production environments.

Experimental results indicate that the proposed model exhibits significant competitive advantages in detection accuracy compared to other algorithms like SSD and Faster R-CNN. When compared to YOLO-series models, the improved model reduces memory consumption and detection time while maintaining high detection rates and precision. Its parameters and computations are 2.32M and 6.4G, respectively, accounting for only 6.353% and 6.201% of YOLOv7's values. Training comparisons on the same dataset between YOLOv8n, YOLOv5n, Fast R-CNN, and SSD show that YOLOv8n achieves the highest mAP@50 (89.52%) with 3.01 GFlops.

However, the developed suboptimal health behavior detection model was validated only at a single experimental site. As chicken behaviors are significantly influenced by environmental variations, this study did not assess the model's generalizability across different settings. Additionally, behavioral preferences vary among chicken breeds, which were not evaluated in this work. Furthermore, the model was designed and optimized for deployment on cage inspection robots, and field testing on such robots will be conducted in future work.

References

[1] National Bureau of Statistics.Livestock product[EB/OL].[2024-03-20]. https://data.stats.gov.cn/easyquery.htm?Cn=C01□zb=A060601□sj= 2023.

[2] Bloch V □ Barchilon N □ Halachmi I □ et al. Automatic broiler temperature measuring by thermal camera[J]. Biosystems Engineering,2020,199: 127-134□

[3] SOZZI M, PILLAN G,ClARELLl C,et al.Measuring comfort behaviorsin laying hens using deep-learning tools[J] .Anima-ls,2022,13(1):33.

[4] Jacob F G, dos Baracho M, de Nääs I, et al. The use of infrared thermography in the identification of pododermatitis in broilers[J]. Engenharia Agrícola, 2016,36(2):253-259.

[5] Chien Y R, Chen Y X. An RFID-based smart nest box: an experimental study of laying performance and behavior of individual hens[J]. Sensors, 2018,18(3) : 859.

[6] LI Q,CHU M,KANG X,et al.Temporal aggregation network using micromotion features for early lameness recognition in dairy cows[J].Computers and Electronics in Agriculture,2023,204:107562.

[7] Amraei S, Abdanan M S, Salari S. Broiler weight estimation based on machine vision and artificial neural network[J]. British Poultry Science, 2017, 58(2): 200-205.

[8] Taylor P S, Hemsworth P H, Groves P J, et al. Ranging behaviour of commercial free-range broiler chickens 2: individual variation[J] Animals, 2017, 7(7): 55.

[9] Fazzari, E., Romano, D., Falchi, F. et al. ARTEMIS: animal recognition through enhanced multimodal integration system[J]. Int. J. Mach. Learn. & Cyber. 16, 5877–5892 (2025).

[10] X. L. Ng, K. E. Ong, Q. Zheng, Y. Ni, S. Y. Yeo and J. Liu, "Animal Kingdom: A Large and Diverse Dataset for Animal Behavior Understanding," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 19001-19012, doi: 10.1109/CVPR52688.2022.01844.

[11] Mortensen A K, Lisouski P, Ahrendt P. Weight prediction of broiler chickens using 3D computer vision[J]. Computers and Electronics in Agriculture, 2016, 123: 319-326.

[12] Johansen S V, Bendtsen J D, R-Jensen M, et al. Broiler weight forecasting using dynamic neural network models with input variable selection[J]. Computers and Electronics in Agriculture, 2019, 159:97-109.

[13] Linhoss J E, Davis J D, Campbell J C, et al. Light intensity and uniformity in commercial broiler houses using lighting programs derived from Global Animal Partnership(GAP) lighting standards[J]. Journal of Applied Poultry Research, 2023, 32(1):100309.

[14] Costantino A, Fabrizio E, Ghiggini A, et al. Climate control in broiler houses: a thermal model for the calculation of the energy use and indoor environmental conditions[J]. Energy and Buildings, 2018, 169: 110-126.

[15] Edoardo Fazzari, Donato Romano, Fabrizio Falchi, Cesare Stefanini,Animal behavior analysis methods using deep learning: A survey,Expert Systems with Applications,Volume 289,2025,128330,ISSN 0957-4174

[16] Mohialdin M A, Elbarrany M A ,Atia A . Chicken Behavior Analysis for Surveillance in Poultry Farms[J].International Journal of Advanced Computer Science and Applications (IJACSA),2023,14(3):

[17] Yao W, Deng W, Xu Y, et al. Poultry sub-health status monitoring and health warning prospect[J]. 2023.

[18] Wu D H,Cui D,Zhou M C,et al.Information perception in modern poultry farming:a review[J].Computers and Electronics in Agriculture,2022,199:107131.

[19] Yang X, Bist R, Paneru B, et al. Monitoring activity index and behaviors of cage-free hens with advanced deep learning technologies[J]. Poultry Science, 2024, 103(11): 104193.

[20] Li, J., et al. (2024). YOLOv8-EMA: Enhanced multi-scale feature fusion for real-time object detection. Pattern Recognition Letters, 178, 45-52. (EMA integration for YOLOv8, focusing on small-object detection)

[21] Zhang, H., et al. (2025). Attention-driven YOLOv9 for fine-grained action recognition in agricultural scenes. Computers and Electronics in Agriculture, 221, 108123. (EMA-based multi-scale feature aggregation for livestock behavior analysis)

[22] Wang, Y., et al. (2023). Efficient multi-scale attention for lightweight YOLO models: Application to edge-device deployment. IEEE Access, 11, 98765-98778. (EMA optimization for resource-constrained YOLO variants)
[23] Süzen A A, Duman B, Şen B. Benchmark analysis of jetson tx2, jetson nano and raspberry pi using deep-cnn[C]//2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE, 2020: 1-5.
[24] https://pypi.org/project/labelImg/
[25] Feroz M A, Sultana M, Hasan M R, et al. Object detection and classification from a real-time video using SSD and YOLO models[M]//Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2021. Singapore: Springer Singapore, 2021: 37-47.
[26] Ilani M A, Banad Y M. LabelImg: CNN-Based Surface Defect Detection[J]. arXiv preprint arXiv:2509.05813, 2025.
[27] J. Li, C. Chen, and H. Wang, "C2f-Net: Cross-level feature fusion for real-time object detection", IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 7, pp. 4561–4574, Jul. 2021.
[28] A. Block and Z. Zhang, "Faster EMA: Accelerated exponential moving average for efficient deep learning training", Adv. Neural Inf. Process. Syst., vol. 36, pp. 12890–12902, 2023.
[29] G. Liu, F. Liu, T. Okada, J. M. Freeman, and G. Sapiro, "Image inpainting for irregular holes using partial convolutions", in Proc. Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, 2018, pp. 85–100.
[30] van Hertem T, Norton T, Berckmans D, et al. Predicting broiler gait scores from activity monitoring and flock data[J]. Biosystems Engineering, 2018, 173: 93-102.
[31] Fazzari E, Romano D, Falchi F, et al. Real-Time Behavior Recognition Using a Legged Robot for Animal–Robot Interaction[J]. Journal of Field Robotics, 2025.
[32] Fazzari E, Romano D, Falchi F, et al. Selective state models are what you need for animal action recognition[J]. Ecological Informatics, 2025, 85: 102955.

Funding

This work was supported by the National Key R&D Program of China (Grant No. 2023YFD2000800).

Author Contributions Statement

Y.T. conducted the experiments, curated the dataset, and led the model development.

J.W. contributed to data annotation, preprocessing, and experimental validation.

B.X. assisted in methodology design and performed comparative analyses.

R.K. contributed to equipment deployment and video acquisition in the poultry houses.

C.Y. supported algorithm implementation and participated in result interpretation.

J.L. contributed to data management, figure preparation, and manuscript editing.

Z.M. assisted in system construction and visualization of detection results.

L.L. supervised the engineering workflow, provided project coordination, and contributed to manuscript revisions.

M.S. conceived and supervised the project, secured funding, and guided the overall research direction.

Data availability

The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.