Response to Reviewer 2 Comments

1. Summary

Thank you for taking the time to carefully review our manuscript. We appreciate your thorough review and the valuable feedback you have provided. Please find the detailed responses and corresponding revisions highlighted in the re-submitted files.

2. Questions for General Evaluation	**Reviewer's Evaluation**	**Response and Revisions**
Does the introduction provide sufficient background and include all relevant references?	Yes	Thank you for your review
Are all the cited references relevant to the research?	Yes	Thank you for your review
Is the research design appropriate?	Yes	Thank you for your review
Are the methods adequately described?	Can be improved	Thank you for your review. We have made corresponding changes in response to your comments, as detailed below
Are the results clearly presented?	Yes	Thank you for your review
Are the conclusions supported by the results?	Yes	Thank you for your review

3. Point-by-point response to Comments and Suggestions for Authors

Comments 1: Term consistency: The prediction classes, real and fake eggs, were not described consistently over the entire manuscript. For example, in Line 50, the eggs to replace the real eggs were called "fake eggs," In Line 126, Caption in Figures 8 and 10, they were called "false eggs." Since "True" and "False" are specific terms in classification problems. Using "False" without referring to a wrong prediction in such a task may introduce confusion to readers. Hence, I suggest using "fake eggs" to minimize the ambiguity.

Response 1: Thank you for pointing this out. We have made targeted revisions to the manuscript, unifying the terms into real pigeon eggs and fake pigeon eggs, and the specific revisions are highlighted in yellow in the attachment.

Comments 2: Lack of details in Table and Figure captions: Although the authors provide abundant information, it successfully makes the work more convincing. Many of the figures and tables lack essential information to be self-explanatory. I have pointed them out specifically in my minor comments.

Minor comments:

- Line 178: Define C2f.
- Line 186: Explain k (kernel size?) and 'cp'.
- Line 189: Explain t.
- Figure 4: No sections (a) and (b) were found in the figure. Also, "h," "w," "k," "cp," "BN" (batch normalization?), and ReLU (not Relu) should be briefly explained in the figure caption. It should be similar to what they were presented in lines 213-215.
- Figure 5: Same to the figure 4. No sections (a) and (b) were found in the figure.
- Figure 6: Explain terms including "gs^2", "sh," "sw," "g," and "o."
- Figure 7: Great figure. It can be more helpful to the reader if the author can present the full names of "CF, "CFE, "SPPF, and "CBS" in the caption.
- Table 2.: "Ir0" should be "lr0" (initial learning rate). The first letter is not correct.
- Table 2.: Including the hyperparameters in the warm-up stage (the first 3 epochs) would be better.
- Table 3.: The modules A to D should be described in the table caption, not only in the manuscript (Lines 333 to 338).
- Table 3.: What is the confidence threshold for the F1 metric?
- Line 371: The space should be moved to after the comma mark: "For the more stringent ," -> "For the more stringent, "
- Figure 9. The meaning of the colors (Line 397) should also go to the figure caption.
- Figure 10. True and false should be real and fake (eggs). Again, using "true" and "false" in a multi-classification problem can be confusing.
- Line 399: Reference for Grad-CAM?
- Line 429. Should be real and fake pigeon eggs.

Response 2: Thank you for pointing this out. The question you raised is outstanding, and it will provide great assistance for our subsequent research. this point. We have made the following changes in response to your suggestion.

Minor comments:	**Response and Revisions:**
Line 178: Define C2f.	We have defined and drawn the structure diagram of C3 and C2f modules in Section 2.2 of the manuscript, and the specific modifications are as follows : It replaces the C3 structure of YOLOv5 with C2f structure. The C2f module can com-bine advanced features with context information to enhance the gradient flow of the model and the feature representation capability of the network by adding additional jump connections, thus improving the detection accuracy. The specific module structure is

	shown in Fig. 4. (This is detailed in lines 167-171 and Figure 4 of the manuscript)
Line 186: Explain k (kernel size?) and 'cp'.	We have added an explanation in line 197 of the manuscript, the specific content is as follows: k represents kernel size, c_p represents the number of PConv channels.
Line 189: Explain t.	Sorry, this is our mistake, we have deleted t in the manuscript
Figure 4: No sections (a) and (b) were found in the figure. Also, "h," "w," "k," "cp," "BN" (batch normalization?), and ReLU (not Relu) should be briefly explained in the figure caption. It should be similar to what they were presented in lines 213-215.	We added (a) and (b) to the figure 4 , modified Relu to ReLU and have added an explanation in lines 193-195 of the manuscript, the specific content is as follows: Here, h, w represents the height and width of the feature graph, BN represents the batch normalization and $ReLU$ represents rectified linear unit.
Figure 5: Same to the figure 4. No sections (a) and (b) were found in the figure.	We added (a) and (b) to the figure 5 and modified Relu to ReLU
Figure 6: Explain terms including "gs^2", "sh," "sw," "g," and "o."	We have explained the terms in the title of the figure, as follows: Figure 7. Dysample network structure: (a) Sampling based dynamic upsampling; (b) Sampling point generator in DySample. The input feature ,upsample feature, generated offset, and original grid are denoted by χ, χ^', G and O, respectively. σ denotes the sigmoid function. sh represents the sampled height, sw represents the sampled width. gs^2 represents the number of channels after the feature graph passes through the linear layer. (Detailed in the manuscript in lines 260-264 and Figure 7)
Figure 7: Great figure. It can be more helpful to the reader if the author can present the full names of "CF, "CFE, "SPPF, and "CBS" in the caption.	We have explained the terms in the title of the figure, as follows: Figure 8. YOLOv8-PG overall framework and its constituent modules: (a)YOLOv8-PG, (b)C2f-Faster, (c)C2f-Faster-EMA, (d)Spatial Pyramid Pooling Fusion, (e)Conv-BN-SiLU.
Table 2.: "Ir0" should be "lr0" (initial learning rate). The first letter is not correct.	We have made corrections in Table 2.

Table 2.: Including the hyperparameters in the warm-up stage (the first 3 epochs) would be better.	We added the specific values of the Warmup epochs and Warmup momentum hyperparameters in Table 2
Table 3.: The modules A to D should be described in the table caption, not only in the manuscript (Lines 333 to 338).	We have added the explanation of A to D in the caption of Table 3, the details are as follows: Note: "√" represents the model's usage of a particular module, while "×" indicates that the model does not utilize that module. A: Replaced C2f with C2f-Faster-EMA in the backbone network; B: Replaced C2f with C2f-Faster in the backbone network; C: Introduce Dysample in the neck network; D: Replaced the loss function with EMASlideLoss.(It's in lines 363-366 of the manuscript)
Table 3.: What is the confidence threshold for the F1 metric?	We provide additional explanations in section 3.2 of the paper: F1 represents the harmonic mean of precision and recall with the confidence thresh-old is 0.5. (It's in lines 342-343 of the manuscript)
Line 371: The space should be moved to after the comma mark: “For the more stringent ,” -> “For the more stringent, “	Thank you for pointing out this problem, and we have modified it
Figure 9. The meaning of the colors (Line 397) should also go to the figure caption.	We explain the colors in the title of Figure 9, as follows: Colors represent scalars of one order of magnitude, with hot tones (such as red or yellow) repre-senting areas of high activity or importance, and cool tones (such as blue or green) representing areas of low activity or importance. (It's in lines 429-432 of the manuscript)
Figure 10. True and false should be real and fake (eggs). Again, using “true” and “false” in a multi-classification problem can be confusing.	Thank you for pointing out this problem, and we have modified it.
Line 399: Reference for Grad-CAM?	Thank you for pointing out this problem, and we have already cited the relevant references. (References 31)
Line 429. Should be real and fake pigeon eggs.	Thank you for pointing out this problem, and we have modified it.

Comments 3: Abbreviation: Many letter abbreviations were presented without explaining their first occurrence.

Response 3: Thank you for pointing out this problem. We have explained the abbreviation when it first appeared in the manuscript and added an extra table for readers to better read our article. The table is as follows:

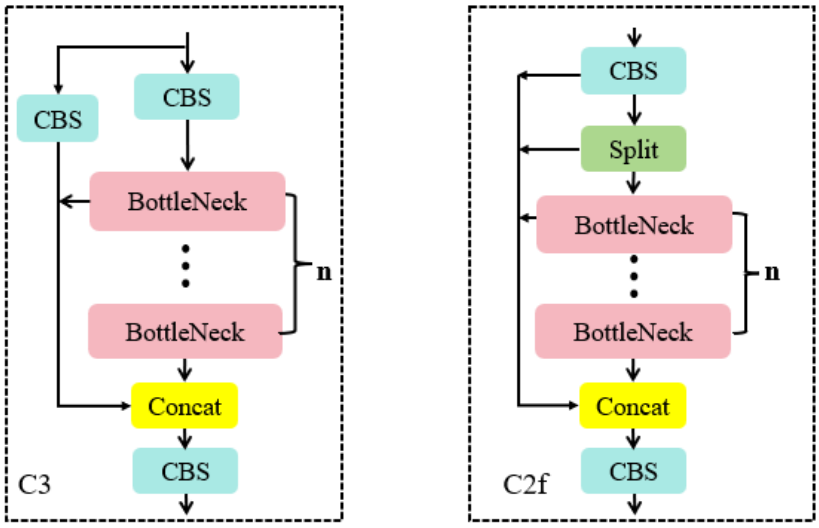
EMA	Efficient Multi-scale Attention
PConv	Partial Convolution
EXPMA	Exponential Moving Average
EMASlideLoss	Exponential Moving Average Slide Loss
Faster R-CNN	Faster Region-based Convolutional Neural Networks
Mask R-CNN	Mask Region-based Convolutional Neural Networks
YOLO	You Only Look Once
MATLAB	Matrix Laboratory
CA	Coordinate Attention
CBAM	Convolutional Block Attention Module
BiFPN	Bidirectional Feature Pyramid Network
SOTA	State Of The Art
ELAN	Efficient Layer Attention Network
CIOU Loss	Complete Intersection over Union Loss
DFL Loss	Distribution Focal Loss
BCE Loss	Binary Cross-Entropy Loss
IoU	Intersection over Union
Conv	Convolution
SiLU	Sigmoid Linear Unit
BN	Batch Normalization
SPPF	Spatial Pyramid Pooling-Fast
GFLOPs	Giga Floating-point Operations Per Second
P	Precision
R	Recall
F1	F1-score
AP	Average Precision
mAP50	mean Average Precision at IoU threshold of 0.5
mAP75	mean Average Precision at IoU threshold of 0.75
mAP50-95	mean Average Precision at a range of IoU thresholds from 0.50 to 0.95
FPS	Frame Per Second

Grad-CAM	Gradient-weighted Class Activation Mapping

Comments 4: Educational standpoints: Since it is an agriculture/biology journal, it is necessary to describe more of the concepts that might be recognized as common sense in the community of computer vision. For example, what are the definitions and functions of the "Input," "Backbone," "Neck," and "Detection Head" network? Also, would it be beneficial if the authors could explain the merit of C2f (YOLOv8) over C3 (YOLOv5) before replacing it with the proposed architecture (YOLOV8-PG)?

Response 4: Thank you for your valuable comments. In section 2.2 of the manuscript, we have added an introduction to the network model structure and explained why C2f is superior to C3 by drawing the structure diagram of C3 and C2f. The details are as follows:

1. The overall structure of the network includes the Input (receives input images and converts them into a format that the model can process), Backbone network (extract the target feature from the image), Neck network (realize image feature fusion), and Detection Head network (the output component of the model, responsible for generating inspection results).
2. It replaces the C3 structure of YOLOv5 with C2f structure. The C2f module can com-bine advanced features with context information to enhance the gradient flow of the model and the feature representation capability of the network by adding additional jump connections, thus improving the detection accuracy. The specific module struc-ture is shown in Fig. 4.



(a) (b)

Figure 4. Partial module structure diagram: (a) C3, (b)C2f.

animals

MDPI

Article

Based on Improved YOLOv8: A Study on Lightweight Pigeon Egg Detection Method

Tao Jiang [1,2], Jie Zhou [1,2], Binbin Xie [1,2], Longshen Liu[1,2,*], Chengyue Ji[1,2], Yao Liu[1,2], Binghan Liu[1], and Bo Zhang[1,2]

[1] College of Artificial Intelligence, Nanjing Agricultural University, Nanjing 210031, China; 9213020127@stu.njau.edu.cn (T.J.); 2022819050@stu.njau.edu.cn (J.Z.); 2022819028@stu.njau.edu.cn (B.X.); 9213020527@stu.njau.edu.cn (C.J.) ; 9213020230@stu.njau.edu.cn (Y.L.);9213020201@stu.njau.edu.cn (B.L.); 2022819049@stu.njau.edu.cn (B.Z.);

[2] Key Laboratory of Breeding Equipment, Ministry of Agriculture and Rural Affairs, Nanjing 210031, China;

* Correspondence: liulongshen@njau.edu.cn;

Simple Summary: The utilization of computer vision technology and automation for monitoring and collecting pigeon eggs is of significant importance for improving labor productivity and the breeding of egg-producing pigeons. Currently, research both domestically and internationally has predominantly focused on the detection of eggs from poultry such as chickens, ducks, and geese, leaving pigeon egg recognition largely unexplored. This study proposes an effective and lightweight network model, YOLOv8-PG, based on YOLOv8n, which maintains high detection accuracy while reducing the model's parameter count and computational load. This approach facilitates cost reduction in deployment and enhances feasibility for implementation on mobile robotic platforms.

Abstract: In response to the high breakage rate of pigeon eggs and the significant labor costs associated with egg-producing pigeon farming, this study proposes an improved YOLOv8-PG (real versus fake pigeon egg detection) model based on YOLOv8n. Specifically, the Bottleneck in the C2f module of the YOLOv8n backbone network and neck network are replaced with Fasternet-EMA Block and Fasternet Block, respectively. The Fasternet Block is designed based on PConv (Partial Convolution) to reduce model parameter count and computational load efficiently. Furthermore, the incorporation of the EMA (Efficient Multi-scale Attention) mechanism helps mitigate interference from complex environments on pigeon egg feature extraction capabilities. Additionally, Dysample, an ultra-lightweight and effective upsampler, is introduced into the neck network to further enhance performance with lower computational overhead. Finally, the EXPMA (Exponential Moving Average) concept is employed to optimize the SlideLoss and propose the EMASlideLoss classification loss function, addressing the issue of imbalanced data samples and enhancing the model's robustness. The experimental results showed that the F1-score, mAP50-95 and mAP75 of YOLOv8-PG increased by 0.76%, 1.56% and 4.45%, respectively, compared with the baseline YOLOv8n model. Moreover, the model's parameter count and computational load are reduced by 24.69% and 22.89%, respectively. Compared to detection models such as Faster R-CNN, YOLOv5s, YOLOv7, and YOLOv8s, YOLOv8-PG exhibits superior performance. Additionally, the reduction in parameter count and computational load contributes to lowering the model deployment costs and facilitates its implementation on mobile robotic platforms.

Keywords: Pigeon egg detection; YOLOv8; Partial Convolution; Efficient multi-scale attention; Exponential Moving Average;

Citation: To be added by editorial staff during production.

Academic Editor: Firstname Lastname

Received: date
Revised: date
Accepted: date
Published: date

Copyright: © 2024 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As of today, pigeon farming has rapidly emerged as an industry, with pigeons now recognized as the fourth major poultry species alongside chickens, ducks, and geese [1]. Pigeon eggs are rich in nutrients such as protein, iron, and calcium, and they are easily

digestible and absorbable. Consuming pigeon eggs can improve skin quality, blood circulation, and possess detoxifying properties, making them a high-quality nutritional product [2,3]. However, pigeon eggshells are relatively soft and prone to breakage from being stepped on or pecked by pigeons. In the actual production and breeding process, when the phenomenon of egg laying is detected, pigeon eggs should be immediately placed into electrical incubation equipment and replaced with fake eggs to minimize the impact on the physiological habits of pigeons, ensuring production efficiency and preventing the occurrence of abandoned hatching [4]. Therefore, rapid and accurate identification of pigeon eggs is particularly important for the management of pigeon eggs.

Currently, both domestically and internationally, there are numerous techniques for detecting various types of poultry eggs using computer vision technology. With the continuous development of technology, certain characteristics and advantages have been formed. In 2008, Pourreza et al. proposed grayscale thresholding of target regions for surface defect detection on poultry eggs [5]. By comparing the ratio of the projected area after thresholding to the target region with a threshold value, they achieved a detection accuracy of 99%. In 2010, Deng et al. used image enhancement algorithms to highlight crack features and applied visual detection to crack egg detection, achieving a high detection accuracy of 98% [6]. In 2012, Lunadei et al. designed a multispectral image processing algorithm to differentiate stains from normal egg colors, with a recognition speed of 50 ms and a recognition rate of 98% [7]. In 2014, Wang extracted 24 physical parameters for crack egg detection, achieving a detection accuracy of 96.67% [8]. In 2017, Sunardi et al. [9] applied smartphones, thermal imaging cameras, and MATLAB for poultry egg recognition, achieving a recognition accuracy of 100%[9]. In the same year, Ang et al. proposed a method combining robots to statistically count and collect eggs in free-range chicken farms, with a positioning error within 2 cm [10]. In 2018, Ab Nasir A et al. designed an automatic egg grading system with positioning and recognition accuracies exceeding 95% [11]. In 2023, Li et al. used an improved YOLOv7 network (MobileOne-YOLO) for detecting fertilized duck eggs, achieving significantly improved FPS performance by 41.6% while maintaining the same accuracy as YOLOv7 [12]. With the development of computer vision and deep learning, the accuracy of many poultry egg detection models has exceeded 95%, and they perform well in complex background environments. However, most methods only focus on accuracy and do not consider model parameter count and computational load.[13]

In 2023, researchers developed machine vision solutions for egg detection on conveyor belts. Huang et al. utilized the CA attention mechanism, BiFPN, and GSConv to enhance YOLOv5 and combined it with byte tracking algorithms to detect broken eggs [14]. On the other hand, Luo et al. improved YOLOv5 using BiFPN and CBAM to detect leaky eggs [15]. Both Huang and Luo et al. employed different approaches to enhance YOLOv5 for detecting egg defects. It is worth noting that their methods were installed above or at the end of the egg conveyor belt. However, pigeon eggs, with lower shell strength and greater fragility compared to eggs from other poultry such as chickens, ducks, and geese, are not typically transported using conveyor belts. Instead, they are inspected and collected manually, leading to significant manual labor and issues such as egg breakage and embryo death due to prolonged exposure in pigeon coops. To enhance the level of intelligence and automation in pigeon egg breeding, and to increase labor productivity, it is crucial to use pigeon egg picking robots for detecting egg laying and collecting pigeon eggs. The key to the development of pigeon egg picking robots lies in the development of precise and efficient pigeon egg detection algorithms.

Currently, research on egg detection has predominantly focused on poultry such as chickens, ducks, and geese, leaving pigeon egg recognition largely unexplored. Moreover, existing deep learning models based on image and video understanding mostly remain in the research stage. Many existing systems rely on cloud computing models, and few scholars have deployed algorithms on embedded devices to accelerate model deployment. Therefore, this study established a comprehensive database of pigeon eggs from Silver

King pigeons, with multiple angles. It distinguished fake pigeon eggs by labeling and specifically adopted C2f-Faster-EMA (CFE) and C2f-Faster (CF) to replace C2f in the backbone and neck networks. Additionally, it introduced the dynamic upsampler Dysample and designed the EMASlideLoss classification loss function to improve the YOLOv8 object detection algorithm. The proposed YOLOv8-PG algorithm model is efficient and lightweight, making it more suitable for deployment on edge devices and pigeon egg picking robots, thereby contributing to the scientific nature of egg breeding processes for pigeons and reducing manual costs. Comparison with other common object detection algorithms in pigeon egg recognition tasks shows that YOLOv8-PG outperforms most models in terms of both accuracy and efficiency.

2. Materials and Methods

2.1. Data Acquisition and Processing

2.1.1. Data Acquisition

From April 1, 2022, to May 15, 2022, video data of pigeon eggs from 150 pairs of breeding pigeons were collected daily. The Silver King pigeon breed was selected as the focus of the study. Custom brackets with Hikvision T12HV3-IA/PoE cameras were mounted on feeding machines in each row of pigeon coops to capture side-view videos of egg laying activities. The collected videos were in RGB format. The speed of the cameras was approximately 1 m/min, synchronized with the speed of the feeding machines. Each pigeon coop had a width of 50 cm. Data collection was conducted for 30 minutes for every batch of 50 pairs of breeding pigeons. A total of 2832 original images were obtained by extracting frames from the collected videos. An illustration of the data collection setup is shown in Fig. 1.

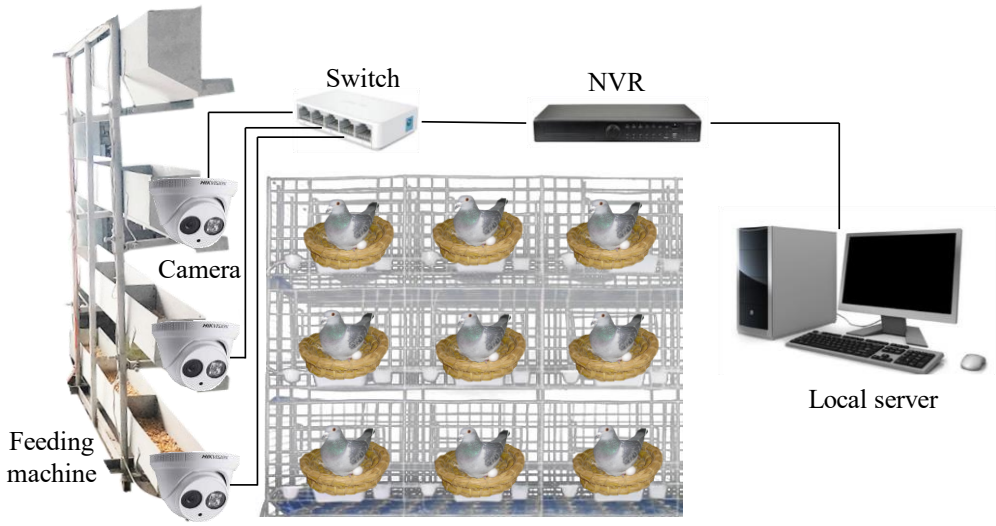


Figure 1. Image data acquisition hardware platform.

2.1.2. Dataset Annotation and Partition

In this study, pigeon eggs were categorized into two categories: real eggs (true) and fake eggs (false). To distinguish fake eggs, they are marked with a black permanent marker to create a cross symbol on their surface. Target extraction of both classes of pigeon eggs is performed on the collected image data, with examples shown in Fig. 2. For the detection of real and fake pigeon eggs, the LabelImg image annotation tool is employed to create the COCO dataset, as depicted in Fig. 3. A total of 2832 image samples were manually annotated. The training, validation, and testing dataset proportions are set at 8:1:2, resulting in dataset sizes of 1982, 283, and 567, respectively.

Annotation guidelines:

1. All annotated bounding boxes should accurately cover the target, closely fitting the target's edges in a minimal rectangular box format [16]. They should not exclude any part of the real or fake pigeon eggs, while also avoiding the inclusion of excessive background information.

2. Annotations for real and fake pigeon eggs should maintain consistency and strictly adhere to the requirements of the pigeon egg schema.

3. For partially occluded pigeon egg targets, the annotation should include the occluding object along with the target, ensuring that no clearly identifiable pigeon egg targets are missed.

4. If pigeon eggs are heavily obscured by pigeon cages, feathers, feces, or other breeding pigeons, making them difficult for the human eye to identify, then those targets should not be annotated.

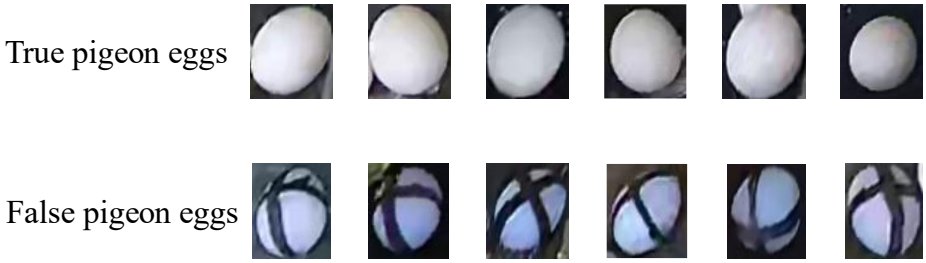


Figure 2. Pigeon egg atlas.

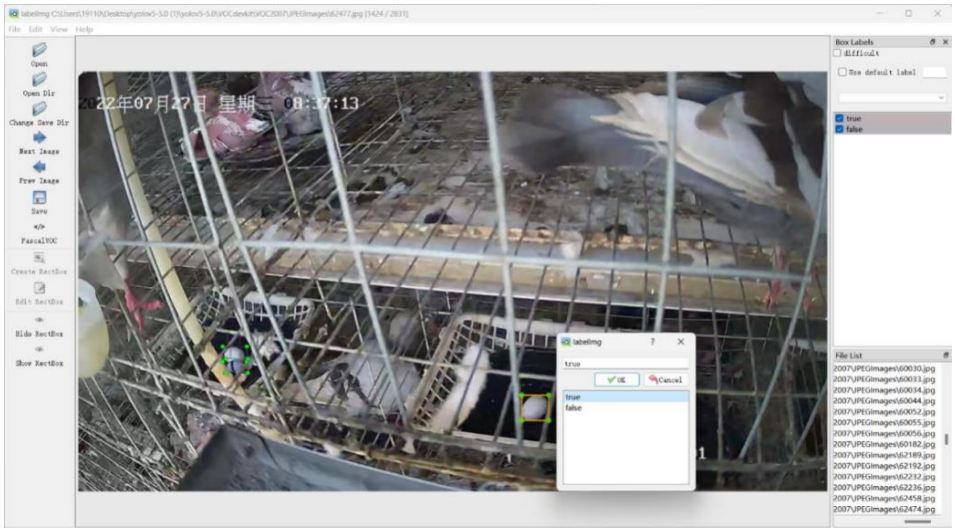


Figure 3. Image annotation interface.

2.2. YOLOv8 network model

The YOLO (You Only Look Once) series algorithms [17–19]are single-stage detection algorithms that balance detection speed with accuracy. YOLOv8, an anchor-free single-stage object detection algorithm, has become one of the mainstream state-of-the-art (SOTA) models. It supports various computer vision tasks such as object detection, instance segmentation, and object tracking. YOLOv8 offers five scaled versions: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. These versions share the same principles but differ in network depth and width. Considering that the output channel numbers vary across different-scale models, a ratio parameter is used to control the channel count. The overall structure of the network includes the Input (receives input images and converts them into a format that the model can process), Backbone network (extract the target feature from

the image), Neck network (realize image feature fusion), and Detection Head network (the output component of the model, responsible for generating inspection results).

It is worth noting that, building upon the success of YOLOv5, YOLOv8 introduces new features and improvements to further enhance its performance and flexibility, including [20]:

1.In the backbone network and neck network, YOLOv8 incorporates the design concept of YOLOv7 ELAN [21]. It replaces the C3 structure of YOLOv5 with C2f structure. The C2f module can combine advanced features with context information to enhance the gradient flow of the model and the feature representation capability of the network by adding additional jump connections, thus improving the detection accuracy. The specific module structure is shown in Fig. 4.

2.YOLOv8 replaces the detection head with a Decoupled-Head structure, separating the classification head from the detection head. Additionally, it switches from Anchor-Based to Anchor-Free detection.

3.Regarding the loss function, YOLOv8 separates the regression and classification tasks in object detection prediction. For the regression task, it employs Distribution Focal Loss (DFL Loss) and Complete Intersection over Union Loss (CIOU Loss). For the classification task, it uses Binary Cross-Entropy Loss (BCE Loss).

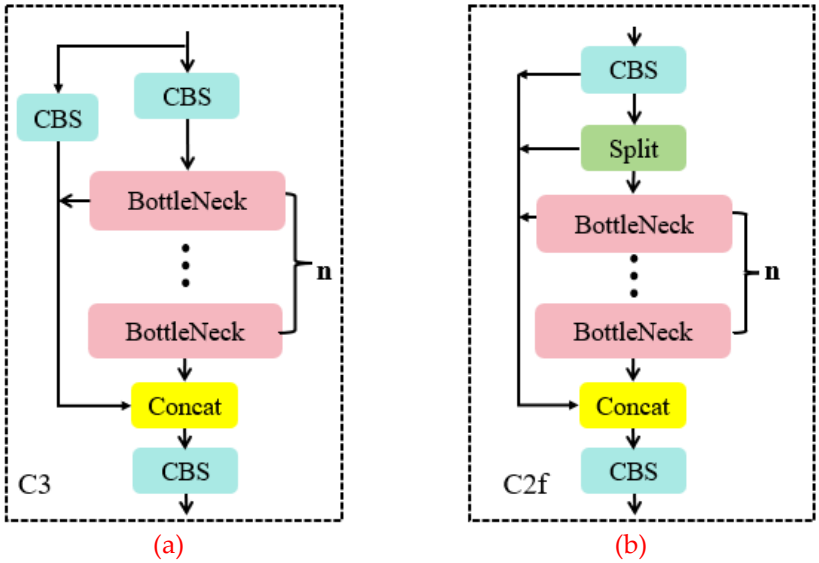


Figure 4. Partial module structure diagram: (a) C3, (b)C2f.

2.3. YOLOv8-PG model improvement strategy

2.3.1. Fasternet Block

To ensure the network has good detection performance, many improvements have focused on reducing the number of floating-point operations (FLOPs). However, the reduction in FLOPs simultaneously leads to frequent memory access by ordinary convolution operations. In this study, we replaced the Bottleneck in C2f with the Fasternet Block [22] to obtain C2f-Faster, thereby significantly reducing the parameter count and computational load of the network model.

The Fasternet Block utilizes a novel convolutional operation called PConv, which effectively extracts spatial features by reducing redundant computations and memory accesses. PConv applies filters only to a subset of input channels for feature extraction while keeping the remaining channels unchanged. Its structure is illustrated in Fig. 5(a). Here, h, w represents the height and width of the feature graph, BN represents the batch normalization and $ReLU$ represents rectified linear unit.

Assuming the input and output feature maps have an equal number of channels, denoted as c, k represents kernel size, c_p represents the number of PConv channels. The FLOPs of PConv are calculated as $h \times w \times k^2 \times c_p^2$. When c_p is $c/4$, the FLOPs of PConv

become 1/16 of those of regular convolution. Additionally, the memory access of PConv is computed as $h \times w \times 2c_p + k^2 \times c_p^2 \approx h \times w \times 2c_p$, which is only a quarter of the regular convolution.

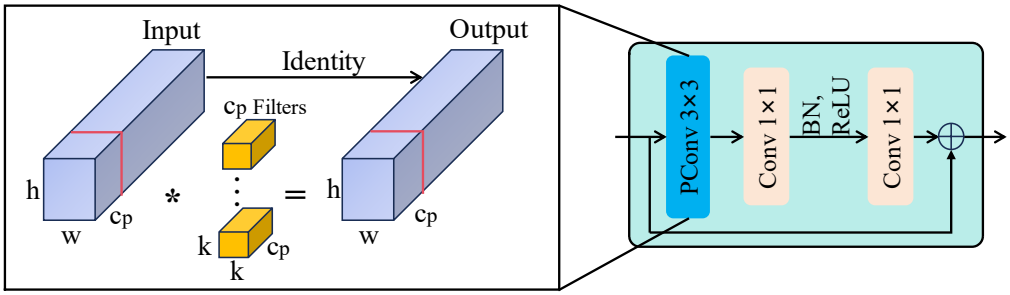


Figure 5. Fasternet Block and its key components: (a) Partial convolution, (b) Fasternet Block.

Fig. 5(b) illustrates the design of the Fasternet Block module. The Fasternet Block module consists of a PConv and two 1 × 1 Conv layers forming a residual block, with shortcut connections included to reuse important features. Normalization layers and activation layers are only applied after the intermediate 1 × 1 Conv layer, aiming to preserve feature diversity and achieve lower latency.

2.3.2. Efficient Multi-scale Attention (EMA)

By incorporating attention mechanisms, algorithms can focus more on key areas or features in images and allocate more attention to them, thereby enhancing the performance of the algorithm in object detection. Due to the presence of dust, feces, feathers, and other debris in pigeon coops, there is significant interference with the detection of pigeon eggs, which prevents the original model from fully extracting features. In this study, the EMA attention mechanism [23,24] is introduced into the YOLOv8n network structure to enhance the model's focus on pigeon egg targets. This enhancement improves the model's ability to extract spatial features more effectively and reduces interference from the complex environment of pigeon coops. Specifically, by leveraging the flexibility and lightweight nature of EMA, it is incorporated into the Fasternet Block to design the Fasternet-EMA Block, as shown in Fig. 6(b).

The EMA module does not perform channel downsampling. Instead, it reshapes the channel dimension into a batch dimension using partial channel dimensions, avoiding downsampling through generalized convolution and preventing loss of feature information. EMA adopts parallel substructures to reduce sequential processing in the network and decrease network depth. The structure of the EMA module is illustrated in Fig. 6(a). Here, h represents the height of the image, w represents the width of the image, c represents the number of channels in the image, and g represents channel grouping.

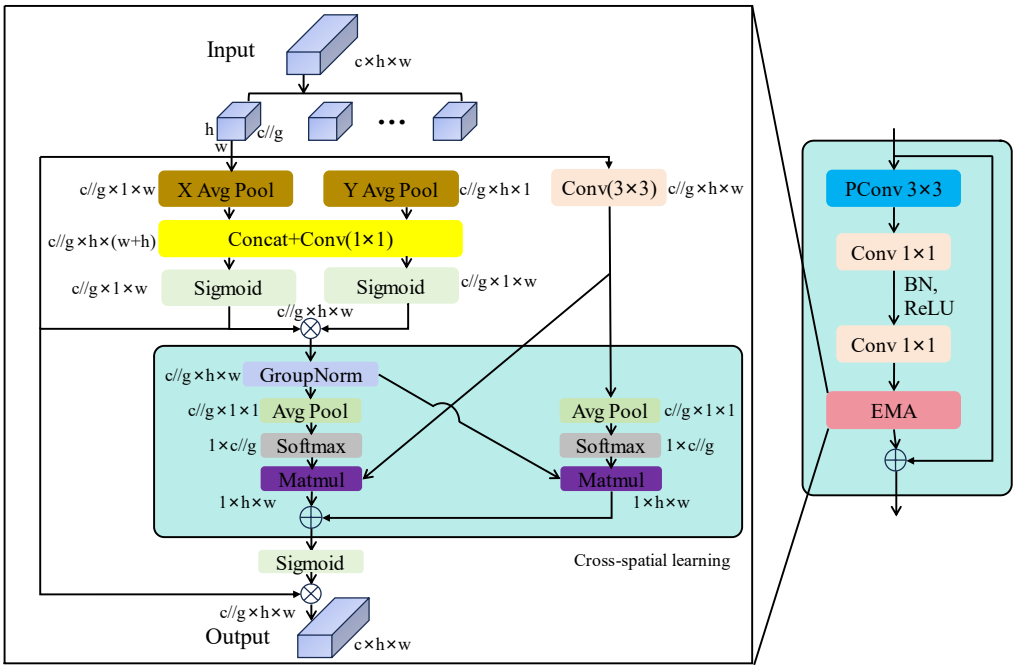


Figure 6. Fasternet-EMA Block and its key components: (a)EMA(b) Fasternet-EMA Block.

Specifically, when presented with specific input feature maps, the EMA attention mechanism initially divides them into G sub-feature maps along the channel dimension to facilitate learning of different semantic features. The EMA module consists of three parallel branches, with two parallel branches located in the 1 × 1 branch and the third branch in the 3 × 3 branch. The 1 × 1 branch utilizes two one-dimensional global average pooling operations to encode channels along two spatial directions, while the 3 × 3 branch stacks single 3 × 3 kernels to capture multi-scale feature representations.

2.3.3. Dysample

In object detection tasks, upsampling operations are required to adjust the size of input feature maps to match the dimensions of the original image, allowing the model to effectively detect objects of various sizes and distances. Traditional upsampling methods typically rely on bilinear interpolation [25]. These methods have inherent limitations and may result in the loss of crucial image details. Moreover, traditional kernel-based upsampling processes entail a significant amount of computation and parameter overhead, which is not conducive to achieving lightweight network architectures [26]. In real-world pigeon coop scenarios, pigeon egg images are relatively small, and issues such as pixel distortion may occur, leading to the loss of fine-grained details and difficulty in learning features during recognition tasks. To address this issue, this paper introduces Dysample [27], a highly lightweight and effective dynamic upsampler, aimed at enhancing the detection capabilities for low-resolution images or smaller pigeon egg targets, while reducing instances of false positives and false negatives. Dysample utilizes a point-based sampling method and a perspective of learning sampling for upsampling, completely avoiding time-consuming dynamic convolution operations and additional sub-networks. It requires fewer computational resources and can enhance image resolution without adding extra burden, thus improving model efficiency and performance with minimal computational cost.

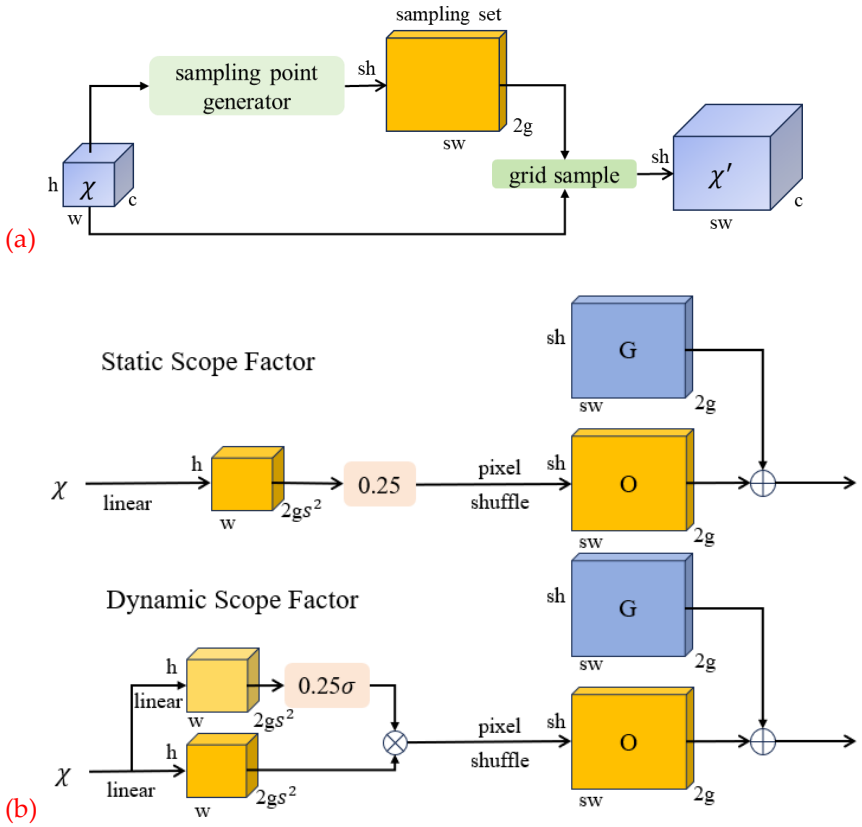


Figure 7. Dysample network structure: (a) Sampling based dynamic upsampling; (b) Sampling point generator in DySample. The input feature ,upsample feature, generated offset, and original grid are denoted by χ, χ', G and O, respectively. σ denotes the sigmoid function. sh represents the sampled height, sw represents the sampled width. gs^2 represents the number of channels after the feature graph passes through the linear layer.

The network structure of Dysample is illustrated in Fig. 7. Its sampling set S consists of the original sampling grid (O) and generated offsets (G). The offsets are generated using a "linear + pixel shuffle" method, where the range of offsets can be determined by static and dynamic factors. Specifically, taking the static factor sampling method as an example, given a feature map of size $c \times h \times w$ and an upsampling factor s, the feature map first passes through a linear layer with input and output channels of c and $2s^2$, respectively. Then, it is reshaped using the pixel shuffle method into $2 \times sh \times sw$, where 2 represents the x and y coordinates. Finally, the upsampled feature map of size $c \times sh \times sw$ can be generated.

2.3.4. EMASlideLoss

The problem of sample imbalance, wherein the quantity of easy samples is often significantly larger than that of difficult samples, has garnered widespread attention. In the SlideLoss loss function [28], the Intersection over Union (IoU) value between predicted boxes and ground truth boxes is utilized as an indicator to distinguish between easy and difficult samples. Due to the limited ability to discern difficult samples, the network model cannot effectively utilize the data during training. The function f(x) is employed to assign a higher weight to difficult samples and a lower weight to easy samples, thereby ensuring that the loss function pays more attention to difficult samples. The allocation rule function is as follows:

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

The function $f(x)$ represents the sliding function operation; x denotes the Intersection over Union (IoU) between predicted boxes and ground truth; μ represents the weight threshold.

In specific, the SlideLoss method utilizes the average IoU value of all bounding boxes as the threshold μ, considering values below μ as negative samples and values above μ as positive samples. In this study, we employ the concept of exponential moving average (EXPMA) [29] to optimize the parameter μ within the model. The specific calculation formula is as follows:

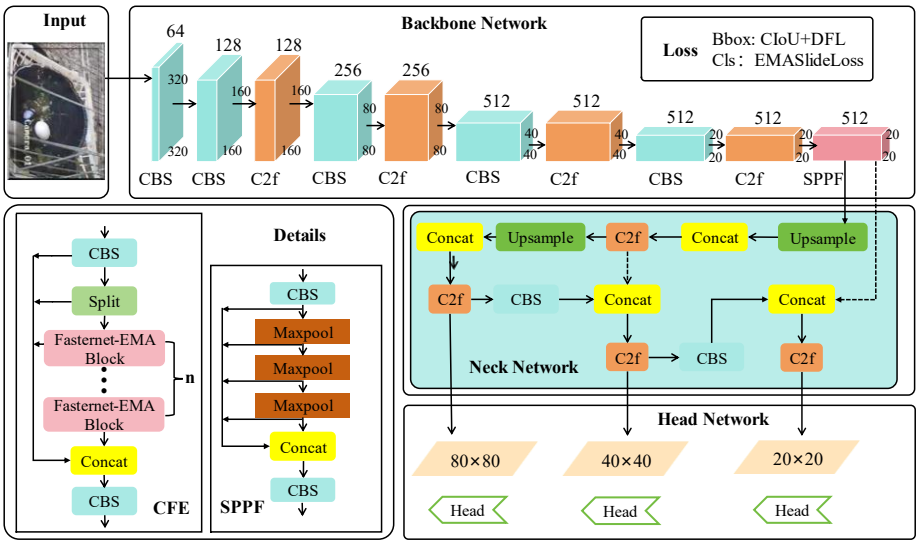
$$\mu_t = \beta \times \mu_{t-1} + (1 - \beta) \times \theta_t \quad (2)$$

Where θ_t represents all parameter weights obtained in the t-th update, μ_t denotes the moving average of all parameters in the t-th update. β denotes the weight parameter.

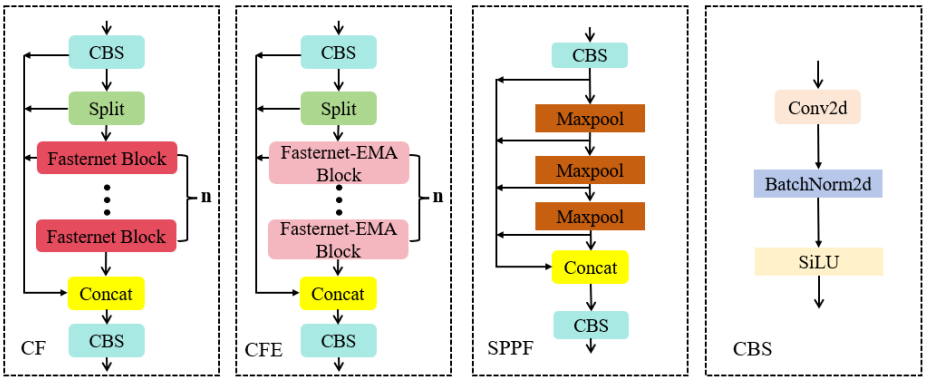
Sliding average can be regarded as the average value of a variable over a certain period of time. Compared to direct assignment, the value obtained through sliding average is smoother and less jittery in the graph, and it does not fluctuate significantly due to occasional abnormal values. Sliding average can enhance the robustness of the model on test data. Although the dataset used in this study is already quite extensive and rich, there is still a shortage in using them to train the model due to the relatively small number of difficult samples. Therefore, this study proposes the EMASlideLoss loss function, which optimizes the SlideLoss function using the exponential moving average (EXPMA) approach to address the issue of sample imbalance and enhance the robustness of the model.

2.4. YOLOv8-PG network model

In summary, this paper has made improvements to the YOLOv8n model in the four aspects mentioned above. The overall network structure of the improved YOLOv8-PG model is illustrated in Fig. 8. Specifically, the C2f-Faster-EMA module was designed for the backbone network, and the C2f-Faster module was used to replace the C2f module in the neck network and introduced the Dysample in the neck network. Regarding the loss function, the EMASlideLoss classification loss function was designed.



(a)



(b) (c) (d) (e)

Figure 8. YOLOv8-PG overall framework and its constituent modules: (a)YOLOv8-PG, (b)C2f-Faster, (c)C2f-Faster-EMA, (d)Spatial Pyramid Pooling Fusion, (e)Conv-BN-SiLU.

3. Experiments and results

3.1. Experimental Details

The model for detecting real and fake pigeon eggs was trained based on the PyTorch framework [30]. The training was conducted for 300 epochs with a batch size of 16. The initial learning rate was set to 0.01, and the weight decay coefficient was set to 0.0005. A warm-up training strategy was employed with warm-up epochs set to 3 and warm-up momentum set to 0.8. To reduce memory usage and improve training speed, mixed precision training strategy was adopted. Additionally, mosaic image augmentation was disabled in the last 10 epochs. For detailed experimental environment and hyperparameter settings, please refer to Table 1 and Table 2, respectively.

Table 1. Experimental environment.

Software environment	Hardware environment
Operating system: Ubuntu20.04	CPU：15 vCPU AMD EPYC 7742 64-Core
Programming language: Python3.8	GPU：RTX 3090(24GB)*1
Deep learning framework: PyTorch2.0.0	
Accelerated environment: CUDA11.8	

Table 2. Hyperparameter Settings.

Parameters	Values
lr0	0.01
lrf	0.01
Momentum	0.937
Weight decay	0.0005
Warmup epochs	3
Warmup momentum	0.8
Batch size	16
Epochs	300
Workers	8

3.2. Model evaluation index

To better evaluate the performance of the model, this experiment adopts the following metrics: F1-score (F1), Mean Average Precision (mAP), model parameters (Params), and Giga Floating Point Operations per second (GFLOPs). The specific calculation formulas are as follows:

$$P = \frac{TP}{TP + FP} \times 100\% \quad (3)$$

$$R = \frac{TP}{TP + FN} \times 100\% \quad (4)$$

$$F1 = \frac{2 \times P \times R}{P + R} \times 100\% \quad (5)$$

$$AP = \int_0^1 P(R)dR \times 100\% \quad (6)$$

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \times 100\% \quad (7)$$

Among them, n index is the number of detection types, and two types of objects are detected in this experiment. TP represents the number of correct detections; FP indicates the number of errors detected. FN indicates the number of missed tests; F1 represents the harmonic mean of precision and recall with the confidence threshold is 0.5. AP represents the accuracy of a certain category; mAP represents the average accuracy for all categories.

The metrics mAP50 and mAP75 correspond to different Intersection over Union (IoU) thresholds. Specifically, mAP50 is calculated using an IoU threshold of 0.50, while mAP75 uses a threshold of 0.75. On the other hand, mAP50-95 refers to the mean average precision calculated over a range of IoU thresholds from 0.50 to 0.95, with increments of 0.05. Generally, the mAP50-95 index is the most stringent as it considers a wider range of IoU thresholds, followed by mAP75, while mAP50 has the lowest threshold and less strict requirements.

3.3. Ablation experiment results

To verify the effectiveness of each improvement on the YOLOv8n algorithm, we conducted ablation experiments on the test set of the same pigeon egg dataset, based on the original YOLOv8n model. The ablation experiments are as follows: A: Replaced C2f with C2f-Faster-EMA in the backbone network to reduce the interference of complex environments on pigeon egg detection. B: Replaced C2f in the neck network with C2f-Faster for lightweight processing. C: Improved the upsampling in the neck network using Dysample to enhance the detection ability of low-resolution and small target pigeon eggs. D: Replaced the loss function with EMASlideLoss to enhance the robustness of the model. The experimental results are shown in Table 3.

Table 3. Comparison results of ablation experiments of different models.

	A	B	C	D	F1	mAP50-95	mAP75	mAP50	Params	GFLOPs
1	×	×	×	×	97.54	68.86	85.01	99.19	3.006M	8.1G
2	√	×	×	×	97.81	69.45	86.67	99.01	2.649M	7.1G
3	√	√	×	×	98.07	69.10	85.97	99.23	2.305M	6.4G
4	√	√	√	×	98.17	69.94	88.57	99.27	2.318M	6.4G
5	√	√	√	√	98.30	70.42	89.46	99.33	2.318M	6.4G

Note: "√" represents the model's usage of a particular module, while "×" indicates that the model does not utilize that module. A: Replaced C2f with C2f-Faster-EMA in the backbone network; B: Replaced C2f with C2f-Faster in the backbone network; C: Introduce Dysample in the neck network; D: Replaced the loss function with EMASlideLoss.

Table 3 shows:

(1) The first group presents the experimental results of the baseline YOLOv8n, serving as the benchmark for the following four groups of experiments. Its F1 is 97.54%, mAP50, mAP75, and mAP50-95 are 99.19%, 85.01%, and 68.86% respectively. The parameter count is 3.006M, and the GFLOPs is 8.1G.

(2) The experiments from the second group to the fifth group progressively incorporate one improvement point at a time. After replacing the Bottleneck of C2f with the Fasternet-EMA Block module in the backbone network of YOLOv8n, the model's accuracy in various aspects has improved, and both the computational complexity and parameter count have decreased. Further introducing Fasternet Block into the neck network C2f results in a slight decrease in accuracy, but the model becomes further lightweight, with a reduction of 0.7G in parameters and 0.344M in computations. After introducing the ultra-lightweight and effective dynamic upsampler Dysample, F1, mAP50-95 and mAP75 have all improved, with negligible impact on parameters and computations. Finally, replacing the EMASlideLoss classification loss function does not increase the model's parameter count or computational complexity but improves the imbalance of samples, leading to a further improvement in model accuracy.

(3) The fifth group shows the results of adding all improvement points. Compared to the baseline model, the YOLOv8-PG model has increased F1 by 0.76% and the mAP50, mAP75, and mAP50-95 metrics have improved by 0.14%, 4.45%, and 1.56% respectively. Additionally, the computational complexity has significantly decreased, with GFLOPs reduced from 8.1G to 6.1G and parameters reduced from 3.006M to 2.318M, representing reductions of 24.69% and 22.89%, respectively.

3.4. Experimental comparison with other models

To validate the superiority of the proposed algorithm, this study conducted comparative experiments with a series of object detection algorithms, including Faster R-CNN, YOLOv5s, YOLOv7, YOLOv8n, and YOLOv8s. The experimental results demonstrate that the proposed algorithm achieves an mAP50 of 99.33%, surpassing Faster R-CNN by 9.86%. For the more stringent ,mAP75 metric, the proposed algorithm outperforms YOLOv5s, YOLOv8n, and YOLOv8s by 1.22%, 4.45%, and 2.06% respectively. Compared to the YOLOv7 model, the proposed algorithm's mAP50-95 shows a slight decrease of 0.26%, but its parameter count and computational load are only 6.353% and 6.201% of the YOLOv7 model, respectively. The comparative experimental results of the algorithms are presented in Table 4.

Table 4. Performance comparison between various network models.

Model	F1	mAP50-95	mAP75	mAP50	Params	GFLOPs
Faster R-CNN	68.98	42.70	32.35	89.47	136.710M	369.7G
YOLOv5s	98.80	69.15	88.24	99.47	7.057M	16.3G
YOLOv7	98.83	70.68	90.07	99.63	36.487M	103.2G
YOLOv8n	97.54	68.86	85.01	99.19	3.006M	8.1G
YOLOv8s	97.63	69.2	87.41	99.25	11.136M	28.4G
YOLOv8-PG	98.3	70.42	89.46	99.33	2.318M	6.4G

The experimental results demonstrate that the optimized YOLOv8n model proposed in this study maintains high detection rates and accuracy while reducing memory overhead and detection time. It consumes fewer memory resources compared to the original YOLOv8 model and mainstream object detection algorithms, making it suitable for deployment on mobile terminals or mobile robotic platforms. Fig. 9 visualizes the detection results of real and fake pigeon eggs for different models.

(a) 

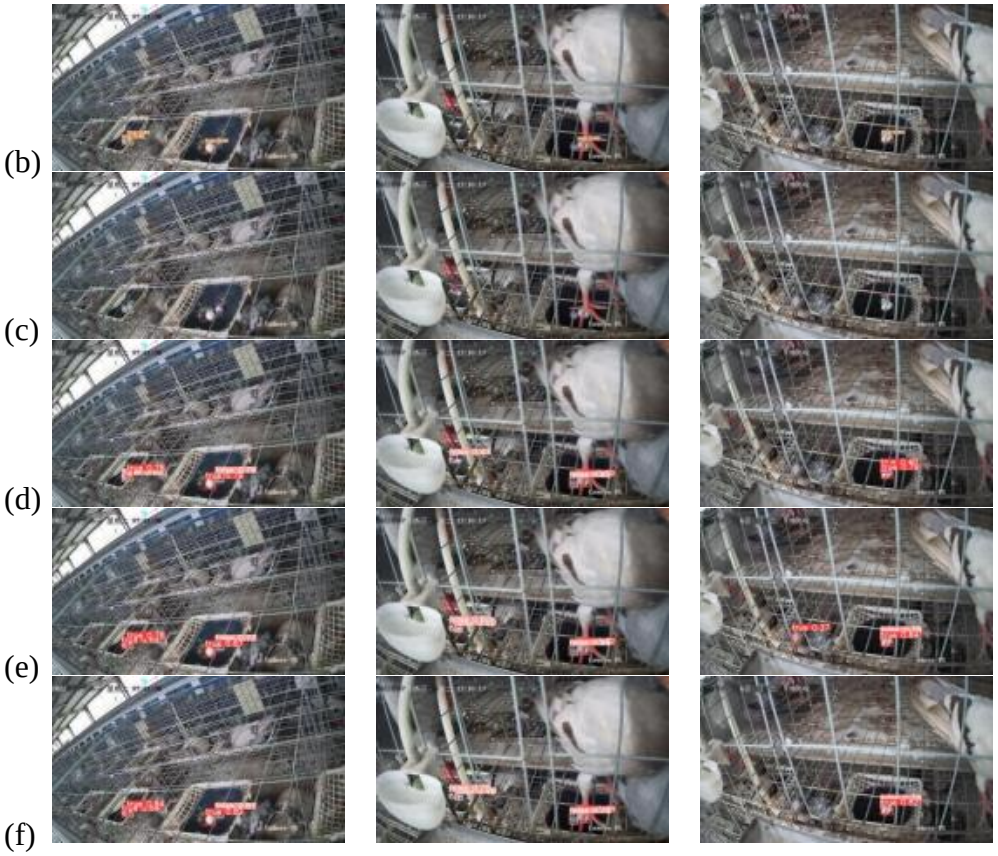


Figure 9. Real and fake pigeon egg detection results of different models:(a)Faster R-CNN (b)YOLOv5s (c)YOLOv7 (d)YOLOv8n (e)YOLOv8s (f) YOLOv8n-PG.

3.5. Model improvement visualization

The heatmap is a visualization technique used in object detection, which can display the distribution of intensity of the objects detected by the model in the input image. Brighter areas indicate higher attention from the model. To visually demonstrate the optimization effect of the proposed YOLOv8-PG model on the real and fake pigeon egg dataset, this study employs the Grad-CAM [31] (Gradient-weighted Class Activation Mapping) algorithm for visual analysis. Partial detection results before and after algorithm improvement are shown in Fig. 10.

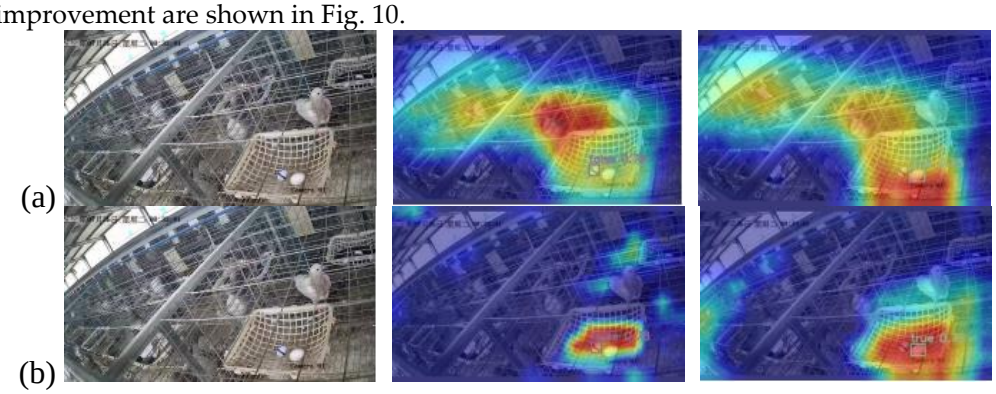


Figure 10. Comparison of thermal map visualization results: (a)YOLOv8n(b)YOLOv8-PG. Colors represent scalars of one order of magnitude, with hot tones (such as red or yellow) representing areas of high activity or importance, and cool tones (such as blue or green) representing areas of low activity or importance.

From Fig. 10, it can be observed that the pigeon farm environment is complex. When YOLOv8n is used for the detection of real and fake pigeon eggs, the background area introduces noise to the model, causing it to focus on some background areas with weaker focus. However, after adding the attention module to improve the model, the noise from the background area is significantly reduced. The model can accurately focus on the pigeon egg area, effectively enhancing the model's ability to extract features of pigeon egg targets in complex environments.

Fig. 11 illustrates the detection effects of YOLOv8n and YOLOv8-PG models on real and fake pigeon eggs under different IOU thresholds, that is, AP values of each category, presented in the form of heat maps. With the increase of IOU, the AP value of each category decreased gradually. When the IOU threshold is lower than 0.8, the AP value of both models is higher than 85% for the real pigeon egg target. The results show that the two models can detect the real pigeon eggs well. For fake pigeon eggs, when the IOU is 0.7 to 0.8, the pigeon eggs are sticky and occluding seriously, and the YOLOv8n model cannot be effectively detected, while the YOLOv8-PG model can reduce the interference of complex environment and the detection accuracy is increased by 2.3%, 9% and 9.42%, respectively.

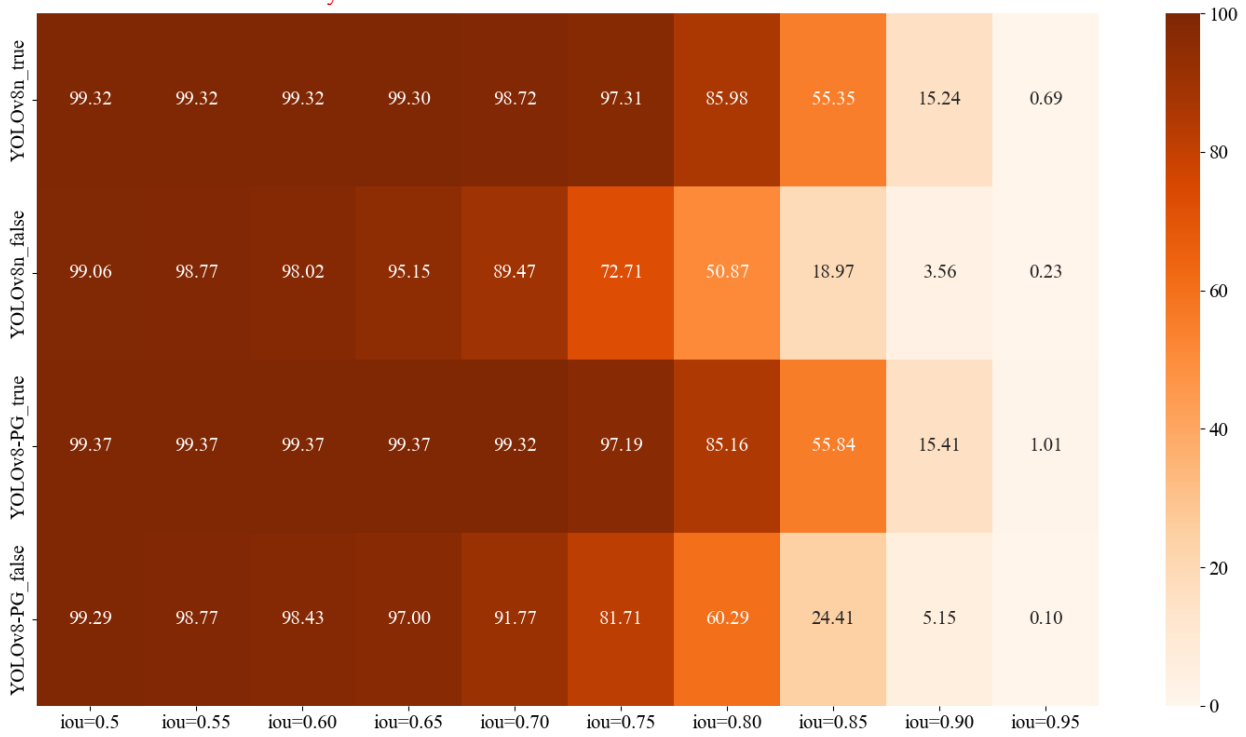


Figure 11. AP value heat map of each detection target

4. Discussion

Since the proposal of detection models based on deep learning, they have been widely applied in various industries, with researchers making great efforts to design suitable models and continuously optimize them. Commonly used object detection methods can be roughly divided into two categories: single-stage detectors and two-stage detectors. Xu et al. [32] deployed an improved Mask R-CNN, a two-stage detector, on mobile robots for egg detection, achieving a high accuracy of 94.18%, but with a slow detection speed of only 0.76 FPS. Due to not requiring a proposal generation stage, single-stage detectors can obtain detection results in a single pass, often resulting in higher processing speeds compared to two-stage detectors. With the iterations of the YOLO series of single-stage detectors, YOLOv8 utilizes anchor-free and decoupled heads to independently process objects, allowing each branch to focus on its own task. YOLOv8 prioritizes the balance between

speed and accuracy, a crucial consideration for integrated system applications. Furthermore, YOLOv8 is more friendly towards detecting small objects, making it the chosen baseline model for our research.

Research has shown that incorporating attention mechanisms and introducing upsamplers [33–36] have been proven to effectively enhance model detection accuracy. Zeng et al. [37] proposed a YOLOv8 model based on the CBAM mechanism, which can effectively select key features of targets, achieving high-precision recognition of coal and gangue. Li et al. [38] proposed an algorithm based on an improved YOLOv5s to achieve target detection and localization in tomato picking. This algorithm replaces upsampling with the CARAFE structure, which improves network sensitivity and accuracy while maintaining lightweightness. It is worth mentioning that different application environments and datasets require customization, modification, and development of models based on single-stage detectors (such as YOLO). Even if the working principle of the model remains unchanged, meaningful architectural modifications must be proposed to perform sufficient customization of deep learning models. Considering the complex environment of pigeon coops and the deployment requirements of models, we improved YOLOv8 using C2f-Faster-EMA, C2f-Faster, and Dysample to reduce the interference of complex environments on the model and enhance the model's ability to detect low-resolution and small targets. Wang et al. [39] addressed the problem of imbalanced difficulty samples by introducing SlideLoss, but they used a fixed value as the threshold for discriminating difficult samples, which cannot improve the model's generalization ability. We optimized the threshold using exponential moving average (EXPMA) and proposed the EMASlideLoss loss function, effectively improving model performance and enhancing model robustness. Combining the above four improvements, this study proposes the YOLOv8-PG model, with F1, mAP50-95, and mAP75 of 98.3%, 70.42%, and 89.46%, respectively, indicating that the model effectively recognizes real and fake pigeon eggs. The parameters and computation amount of the model are 2.318M and 6.4G, respectively, indicating that the model architecture is lightweight and has the potential to be deployed on embedded devices or mobile platforms.

In recent years, scholars have conducted research on model deployment. Ju et al. [40] proposed the MW-YOLOv5s rice recognition model and successfully deployed it on a weeding robot, meeting the practical requirements for both detection accuracy and speed. Yu et al. [41] utilized strategies such as SPD-Conv, CARAFE, and GSConv to propose the lightweight model SOD-YOLOv5n, with a model size of only 3.64M. They successfully deployed the model on Android devices for real-time detection and counting of winter jujubes. In the future, we plan to deploy the YOLOv8-PG model on robots for automated egg detection. However, the model still has some limitations. Firstly, the data collection method in this study involved installing cameras on feeders, resulting in a high installation height and wide field of view. However, the actual perspective of future intelligent egg-picking robots may be limited and more prone to occlusion issues. Secondly, the real and fake pigeon egg dataset in this study only collected images of pigeon eggs from the Silver King breed, which may introduce biases compared to eggs from other breeds. Therefore, in future research, we will further expand the dataset by adding images of pigeon eggs from different angles and different breeds to enhance the model's robustness. Additionally, to deploy the trained model on actual robots, this study will consider converting the trained PyTorch format weights to formats such as ONNX and TensorRT. Furthermore, we will utilize TensorRT for acceleration and test the model's performance on edge devices such as Jetson and Raspberry Pi using methods such as FP16 and INT8 quantization.

5. Conclusions

This paper addresses the high rate of pigeon egg breakage and the high cost of manual labor in pigeon egg breeding by proposing a pigeon egg detection algorithm called YOLOv8-PG based on YOLOv8 design. This algorithm is capable of detection in complex

environments and limited computational resources. Firstly, by combining the Fasternet Block with the EMA attention mechanism and introducing them into the backbone network of the algorithm, the model's feature extraction capability for pigeon egg targets is enhanced. Secondly, introducing C2f-Faster into the neck network of the algorithm further lightweightens the model, reducing the model's parameter and computational complexity. Additionally, the dynamic upsampler Dysample, based on point sampling, is introduced into the model to enhance its detection capability for low-resolution and small-target objects with minimal computational overhead. Finally, the SlideLoss loss function is optimized using the EXPMA concept, and EMASlideLoss is proposed to address the problem of sample imbalance, enhance model robustness, and improve algorithm performance.

The experimental results demonstrate that compared to the two-stage algorithm Faster R-CNN, the model designed in this study shows significant competitiveness in terms of detection accuracy. Compared to other YOLO algorithms in the same series, YOLOv8-PG reduces memory overhead and detection time while maintaining high detection rate and accuracy. Relative to the baseline YOLOv8n the YOLOv8-PG model shows improvements in F1-score by 0.76%, and mAP50, mAP75 and mAP50-95 metrics by 0.14%, 4.45%, and 1.56% respectively. Additionally, there is a significant reduction in computational complexity, with GFLOPs decreasing from 8.1G to 6.1G and parameters from 3.006 M to 2.318 M, representing reductions of 24.69% and 22.89% respectively. The improved YOLOv8-PG model, incorporating various methods, demonstrates potential for integration into pigeon egg picking robots, with satisfactory robustness and applicability in terms of detection accuracy and model deployment feasibility.

Author Contributions: Conceptualization, B.X.; Data curation, T.J. and J. Z.; Formal analysis, Y.L.; Funding acquisition, L.L.; Investigation, B.X. and C.J.; Methodology, T.J., J.Z. and L.L.; Project administration, L.L.; Resources, L.L.; Software, B.X., J.Z. and B.L.; Validation, B.Z.; Visualization, C.J.; Writing – original draft, T.J., B.Z., Y.L.; Writing – review & editing, T.J., J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Science and Technology Program of Liuhe District, Nanjing, Jiangsu Province, China (Project No. LHZC2021N04) and National Student Innovation Research and Entrepreneurship Training , China (Project No. 202310307074Z).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to thank the breeding farm that allowed us to conduct this research in their facilities and generously assisted us.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations:

The following abbreviations are used in this manuscript:

EMA	Efficient Multi-scale Attention
PConv	Partial Convolution
EXPMA	Exponential Moving Average
EMASlideLoss	Exponential Moving Average Slide Loss
Faster R-CNN	Faster Region-based Convolutional Neural Networks
Mask R-CNN	Mask Region-based Convolutional Neural Networks
YOLO	You Only Look Once
MATLAB	Matrix Laboratory

CA	Coordinate Attention
CBAM	Convolutional Block Attention Module
BiFPN	Bidirectional Feature Pyramid Network
SSOTA	state-of-the-art
ELAN	Efficient Layer Attention Network
CIOU Loss	Complete Intersection over Union Loss
DFL Loss	Distribution Focal Loss
BCE Loss	Binary Cross-Entropy Loss
IoU	Intersection over Union
Conv	Convolution
SiLU	Sigmoid Linear Unit
BN	Batch Normalization
SPPF	Spatial Pyramid Pooling-Fast
FLOPs	floating point operations
GFLOPs	Giga Floating-point Operations Per Second
P	Precision
R	Recall
F1	F1-score
AP	Average Precision
mAP50	mean Average Precision at IoU threshold of 0.5
mAP75	mean Average Precision at IoU threshold of 0.75
mAP50-95	mean Average Precision at a range of IoU thresholds from 0.50 to 0.95
FPS	Frame Per Second
Grad-CAM	Gradient-weighted Class Activation Mapping

References

1. Ji, F.; Zhang, D.; Shao, Y.; Yu, X.; Liu, X.; Shan, D.; Wang, Z. Changes in the Diversity and Composition of Gut Microbiota in Pigeon Squabs Infected with Trichomonas Gallinae. *Sci Rep* **2020**, *10*, doi:10.1038/s41598-020-76821-9.
2. Pomianowski, J.F.; Mikulski, D.; Pudyszak, K.; Cooper, R.G.; Angowski, M.; Jóźwik, A.; Horbańczuk, J.O. Chemical Composition, Cholesterol Content, and Fatty Acid Profile of Pigeon Meat as Influenced by Meat-Type Breeds. *Poult Sci* **2009**, *88*, 1306–1309, doi:10.3382/ps.2008-00217.
3. Chang, L.; Zhang, R.; Fu, S.; Mu, C.; Tang, Q.; Bu, Z. Effects of Different Dietary Calcium Levels on the Performance, Egg Quality, and Albumen Transparency of Laying Pigeons. *Animals* **2019**, *9*, doi:10.3390/ani9030110.
4. Adawy, A.M.G.; Abdel-Wareth, A.A.A. Productive Performance and Nutritional of Domesticated Pigeons. Present Status and Future Concerns. *SVU-International Journal of Agricultural Sciences* **2023**, *5*, 160–167, doi:10.21608/svuijas.2023.226150.1305.
5. Pourreza, H.R.; Pourreza, H.R.; Pourreza, R.S.; Fazeli, S.; Taghizadeh, B. Automatic Detection of Eggshell Defects Based on Machine Vision. *Article in Journal of Animal and Veterinary Advances* **2008**, *7*, 1200–1203.

6. Deng, X.; Wang, Q.; Chen, H.; Xie, H. Eggshell Crack Detection Using a Wavelet-Based Support Vector Machine. *Comput Electron Agric* **2010**, *70*, 135–143, doi:10.1016/j.compag.2009.09.016.
7. Lunadei, L.; Ruiz-Garcia, L.; Bodria, L.; Guidetti, R. Automatic Identification of Defects on Eggshell Through a Multispectral Vision System. *Food Bioproc Tech* **2012**, *5*, 3042–3050, doi:10.1007/s11947-011-0672-x.
8. Wang, Y. *Research on the Computer Vision Cracked Eggs Detecting Method*; 2014; Vol. 50;.
9. Sunardi, S.; Yudhana, A.; Saifullah, S. Identity Analysis of Egg Based on Digital and Thermal Imaging: Image Processing and Counting Object Concept. *Article in International Journal of Electrical and Computer Engineering* **2017**, *7*, 200–208, doi:10.11591/ijece.v7i1.12718.
10. IEEE Malaysia Section; IEEE Region 10.; Institute of Electrical and Electronics Engineers *Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017.*; ISBN 9781509011346.
11. Ab Nasir, A.F.; Sabarudin, S.S.; Abdul Majeed, A.P.P.; Abdul Ghani, A.S. Automated Egg Grading System Using Computer Vision: Investigation on Weight Measure versus Shape Parameters. In Proceedings of the IOP Conference Series: Materials Science and Engineering; Institute of Physics Publishing, April 6 2018; Vol. 342.
12. Li, Q.; Shao, Z.; Zhou, W.; Su, Q.; Wang, Q. MobileOne-YOLO: Improving the YOLOv7 Network for the Detection of Unfertilized Duck Eggs and Early Duck Embryo Development - a Novel Approach. *Comput Electron Agric* **2023**, *214*, doi:10.1016/j.compag.2023.108316.
13. Chen, J.; Ma, A.; Huang, L.; Li, H.; Zhang, H.; Huang, Y.; Zhu, T. Efficient and Lightweight Grape and Picking Point Synchronous Detection Model Based on Key Point Detection. *Comput Electron Agric* **2024**, *217*, doi:10.1016/j.compag.2024.108612.
14. Huang, Y.; Luo, Y.; Cao, Y.; Lin, X.; Wei, H.; Wu, M.; Yang, X.; Zhao, Z. Damage Detection of Unwashed Eggs through Video and Deep Learning. *Foods* **2023**, *12*, doi:10.3390/foods12112179.
15. Luo, Y.; Huang, Y.; Wang, Q.; Yuan, K.; Zhao, Z.; Li, Y. An Improved YOLOv5 Model: Application to Leaky Eggs Detection. *LWT* **2023**, *187*, doi:10.1016/j.lwt.2023.115313.
16. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. *You Only Look Once: Unified, Real-Time Object Detection*;
17. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. **2018**.
18. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. **2020**.
19. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. **2016**.
20. Terven, J.; Cordova-Esparza, D. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. **2023**, doi:10.3390/make5040083.
21. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. **2022**.
22. Chen, J.; Kao, S.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; Chan, S.-H.G. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. **2023**.
23. 22.Efficient Multi-Scale Attention Module with Cross-Spatial LearningEMA论文出处.
24. Chen, S.; Li, Y.; Zhang, Y.; Yang, Y.; Zhang, X. Soft X-Ray Image Recognition and Classification of Maize Seed Cracks Based on Image Enhancement and Optimized YOLOv8 Model. *Comput Electron Agric* **2024**, *216*, doi:10.1016/j.compag.2023.108475.
25. Li, Y.; Ma, C.; Li, L.; Wang, R.; Liu, Z.; Sun, Z. Lightweight Tunnel Obstacle Detection Based on Improved YOLOv5. *Sensors* **2024**, *24*, doi:10.3390/s24020395.
26. Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C.C.; Lin, D. CARAFE: Content-Aware ReAssembly of FEatures. **2019**.
27. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to Upsample by Learning to Sample. **2023**.

28. Yu, Z.; Huang, H.; Chen, W.; Su, Y.; Liu, Y.; Wang, X. YOLO-FaceV2: A Scale and Occlusion Aware Face Detector. **2022**.
29. Tarvainen, A.; Valpola, H. Mean Teachers Are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-Supervised Deep Learning Results. **2017**.
30. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury Google, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*; 2019;
31. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. **2016**, doi:10.1007/s11263-019-01228-7.
32. Xu, Z.; Fan, M.; Zhang, Y. Visual Detection of Eggs Based on Deep Learning for Egg Picking Robot. In Proceedings of the Journal of Physics: Conference Series; IOP Publishing Ltd, July 28 2021; Vol. 1976.
33. Qi, J.; Liu, X.; Liu, K.; Xu, F.; Guo, H.; Tian, X.; Li, M.; Bao, Z.; Li, Y. An Improved YOLOv5 Model Based on Visual Attention Mechanism: Application to Recognition of Tomato Virus Disease. *Comput Electron Agric* **2022**, *194*, doi:10.1016/j.compag.2022.106780.
34. Xie, X.; Qin, Y.; Zhang, Z.; Yan, Z.; Jin, H.; Xu, M.; Zhang, C. GY-SLAM: A Dense Semantic SLAM System for Plant Factory Transport Robots. *Sensors* **2024**, *24*, doi:10.3390/s24051374.
35. Huang, D.; Tu, Y.; Zhang, Z.; Ye, Z. A Lightweight Vehicle Detection Method Fusing GSConv and Coordinate Attention Mechanism. *Sensors* **2024**, *24*, 2394, doi:10.3390/s24082394.
36. Jia, Y.; Fu, K.; Lan, H.; Wang, X.; Su, Z. Maize Tassel Detection with CA-YOLO for UAV Images in Complex Field Environments. *Comput Electron Agric* **2024**, *217*, doi:10.1016/j.compag.2023.108562.
37. Zeng, Q.; Zhou, G.; Wan, L.; Wang, L.; Xuan, G.; Shao, Y. Detection of Coal and Gangue Based on Improved YOLOv8. *Sensors* **2024**, *24*, doi:10.3390/s24041246.
38. Li, T.; Sun, M.; He, Q.; Zhang, G.; Shi, G.; Ding, X.; Lin, S. Tomato Recognition and Location Algorithm Based on Improved YOLOv5. *Comput Electron Agric* **2023**, *208*, doi:10.1016/j.compag.2023.107759.
39. Wang, Z.; Wang, X.; Shi, Y.; Qi, H.; Jia, M.; Wang, W. Lightweight Detection Method for X-Ray Security Inspection with Occlusion. *Sensors* **2024**, *24*, doi:10.3390/s24031002.
40. Ju, J.; Chen, G.; Lv, Z.; Zhao, M.; Sun, L.; Wang, Z.; Wang, J. Design and Experiment of an Adaptive Cruise Weeding Robot for Paddy Fields Based on Improved YOLOv5. *Comput Electron Agric* **2024**, *219*, doi:10.1016/j.compag.2024.108824.
41. Yu, C.; Feng, J.; Zheng, Z.; Guo, J.; Hu, Y. A Lightweight SOD-YOLOv5n Model-Based Winter Jujube Detection and Counting Method Deployed on Android. *Comput Electron Agric* **2024**, *218*, doi:10.1016/j.compag.2024.108701.