agriculture

MDPI

Article

DualHet-YOLO: A Dual-Backbone Heterogeneous YOLO Network for Inspection Robots to Recognize Yellow-Feathered Chicken Behavior in Floor-Raised House

Yaobo Zhang, Linwei Chen, Hongfei Chen, Tao Liu, Jinlin Liu, Qiuhong Zhang, Mingduo Yan, Kaiyue Zhao, Shixiu Zhang and Xiuguo Zou *

College of Artificial Intelligence, Nanjing Agricultural University, Nanjing 210031, China; 9223020522@stu.njau.edu.cn (Y.Z.); 9213020625@stu.njau.edu.cn (L.C.); 2023119023@stu.njau.edu.cn (H.C.); 2023819084@stu.njau.edu.cn (T.L.); 2023819082@stu.njau.edu.cn (J.L.); 2023119024@stu.njau.edu.cn (Q.Z.); 9213021013@stu.njau.edu.cn (M.Y.); 9213020111@stu.njau.edu.cn (K.Z.); 1996043@njau.edu.cn (S.Z.)
* Correspondence: zouxiuguo@njau.edu.cn; Tel.: +86-25-58606585

Abstract

The behavior of floor-raised chickens is closely linked to their health status and environmental comfort. As a type of broiler chicken with special behaviors, understanding the daily actions of yellow-feathered chickens is crucial for accurately checking their health and improving breeding practices. Addressing the challenges of high computational complexity and insufficient detection accuracy in existing floor-raised chicken behavior recognition models, a lightweight behavior recognition model was proposed for floor-raised yellow-feathered chickens, based on a Dual-Backbone Heterogeneous YOLO Network. Firstly, DualHet-YOLO enhances the feature extraction capability of floor-raised chicken images through a dual-path feature map extraction architecture and optimizes the localization and classification of multi-scale targets using a TriAxis Unified Detection Head. Secondly, a Proportional Scale IoU loss function is introduced that improves regression accuracy. Finally, a lightweight structure Eff-HetKConv was designed, significantly reducing model parameters and computational complexity. Experiments on a private floor-raised chicken behavior dataset show that, compared with the baseline YOLOv11 model, the DualHet-YOLO model increases the mAP for recognizing five behaviors—pecking, resting, walking, dead, and inactive—from 77.5% to 84.1%. Meanwhile, it reduces model parameters by 14.6% and computational complexity by 29.2%, achieving a synergistic optimization of accuracy and efficiency. This approach provides an effective solution for lightweight object detection in poultry behavior recognition.

Keywords: floor-raised chicken behavior recognition; model lightweighting; YOLO-based target detection; dual-path feature map extraction; TriAxis unified detection head

check for updates

Academic Editor: Luís Manuel Navas Gracia

Received: 11 June 2025
Revised: 4 July 2025
Accepted: 8 July 2025
Published: 12 July 2025

Citation: Zhang, Y.; Chen, L.; Chen, H.; Liu, T.; Liu, J.; Zhang, Q.; Yan, M.; Zhao, K.; Zhang, S.; Zou, X. DualHet-YOLO: A Dual-Backbone Heterogeneous YOLO Network for Inspection Robots to Recognize Yellow-Feathered Chicken Behavior in Floor-Raised House. *Agriculture* **2025**, *15*, 1504. https://doi.org/10.3390/agriculture15141504

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

1.1. Background and Significance

With the deepening of animal welfare concepts and the emergence of consumption upgrading trends, the ecological floor-raised farming model has assumed an increasingly important role in modern agriculture, driven by its natural breeding environment and high-quality product characteristics. Compared with traditional cage systems, floor-raised systems not only provide poultry with activity spaces that cater to their innate behaviors

(such as free movement and dust bathing) but also significantly enhance the sensory quality and nutritional value of poultry products through environmental control [1].

The behavior of floor-raised chickens is closely tied to welfare indicators such as health status and environmental comfort, serving as a critical metric for evaluating livestock and poultry welfare. By recognizing and monitoring their behaviors [2], precise flock management can be achieved, enabling timely detection of diseases and abnormal behaviors [3] to reduce economic losses. For example, abnormal behaviors like feather pecking or fighting often signal the onset of diseases or stress responses [4,5]; behavior recognition technology can identify these anomalies early, allowing for proactive prevention and treatment to avoid large-scale disease outbreaks. In broiler farming, health monitoring currently relies primarily on manual observation, where staff identify health anomalies through behavioral cues. However, this method has significant limitations. First, it requires continuous observation of specific behaviors to ensure accuracy [6]. Second, high labor intensity and subjective judgment variability lead to low monitoring efficiency. Notably, commercial animal behavior analysis systems like EthoVision XT 10 [7] and eYeNamic [8–10] suffer from high cost, functional limitations, and homogeneous analysis targets that struggle to meet the needs of diverse sample analysis in research scenarios. Therefore, developing real-time automated non-destructive monitoring systems is crucial for precise poultry behavior analysis, improving animal welfare, and establishing early warning mechanisms for sick chickens [11].

1.2. Related Works

Recent years have seen growing research in behavior detection. The research focused on behavioral differences between free-range laying hens indoors and outdoors [12]. Zhuang et al. [13] used SSD [14] for sick chicken detection, while Aydin studied chicken lameness detection [15], and E. Tullo focused on leg disease detection [16]. Nasiri et al. proposed a pose estimation model using deep CNNs to detect key body points in walking broilers, feeding sequential points into an LSTM model [17]. In terms of poultry behavior recognition models, Fang et al. combined key point-based neural networks with Naive Bayes Model (NBM)-based classification networks to analyze six broiler behaviors [18]. Pu proposed an automatic CNN method integrated with Kinect sensors to recognize chicken behaviors in farms [19]. Fréjus et al, on the other hand, proposed an automatic recurrent neural network based on attentive chickens for the recognition of laying hen behavior [20], and Guo et al. designed a CNN model for chicken behavior monitoring [21]. YOLO [22] models have also been widely applied in poultry behavior detection. Xie et al. integrated deep learning and pose detection to build a real-time chicken pose detection system based on YOLOv4 [23]. Wang et al. used YOLOv5 for real-time laying hen behavior recognition [24], and Zhang et al. improved the SPPF and C3 modules of YOLOv5s to develop a new model for daily behavior recognition of caged ducks [25]. Yang, X. et al. combined YOLOv8 with DeepSORT, achieving high performance in poultry behavior detection [26].

Some studies have considered the impact of different distribution areas [27] and feeding systems [28,29] on poultry behavior, while others highlight the significance of behavior analysis for animal welfare. The latest definition of animal welfare emphasizes not only positive physical and mental states but also meeting physiological and behavioral needs [30]. However, existing research in floor-raised chicken behavior recognition faces multiple technical bottlenecks:

(1) Multi-scale target challenge: Dense flock distribution and varying distances from cameras lead to a power-law distribution of object scales, making traditional single-scale feature pyramids ineffective for cross-level detection consistency.

(2) Complex environmental interference: Dynamic lighting, vegetation occlusion, and flock overlap in free-range environments significantly reduce the accuracy of conventional detection models.
(3) Hardware deployment constraints: Limited computational resources in edge devices require models to balance real-time performance with parameter efficiency.

1.3. Main Contribution

To address these issues, this study innovatively constructs a Dual-Backbone Heterogeneous YOLOv11 (DualHet-YOLO) network. https://github.com/Yaobo-hub/DualHet (accessed on 30 June 2025). The main contributions are as follows:

(1) Dual-path feature extraction architecture: By using two parallel backbones to extract multi-level and semantic features, and introducing routing and fusion layers for information interaction, this approach effectively solves problems of multi-scale targets and complex backgrounds, significantly improving behavior recognition accuracy.
(2) Eff-HetKConv lightweight structure: Designed to address edge device resource limitations, this structure uses a hybrid configuration of 3×3 and 1×1 heterogeneous convolutional kernels to reduce computations while enhancing feature extraction. Model parameters are reduced from 4.07 million to 3.47 million (14.6% reduction) with improved detection accuracy.
(3) Proportional Scale IoU (Pro-Scale IoU) loss function: By incorporating aspect ratio coefficients and scale-adaptive factors, this function dynamically adjusts regression weights based on target box aspect ratios and sizes, prioritizing long-edge correction and small-object localization to improve positioning accuracy and reduce misdetections/omissions.
(4) TriAxis Unified Detection Head: Equipped with spatial adaptive fusion and channel attention mechanisms, this head adapts to environmental changes by dynamically adjusting feature fusion and enhancing key features, improving the model's robustness in complex backgrounds.

This paper is structured as follows: Section 2 details the dataset and architectural design of DualHet-YOLO. Section 3 evaluates its performance through generalization, comparison, and ablation experiments in complex farm scenarios. Section 4 discusses the proposed method, and the conclusion summarizes key findings and contributions.

2. Materials and Methods

2.1. Data Acquisition and Processing

The data for this study were sourced from a large-scale farm raising floor-raised yellow-feathered broilers in Jinniuhu Street, Liuhe District, Nanjing City, Jiangsu Province, China (118°52′38″ E, 32°26′54″ N). Each chicken coop on the farm measures 5.0 m in width and 12.5 m in length, with an internal area of 62.5 m^2, housing approximately 1258 yellow-feathered broilers. In typical large-scale farms, farmers often exceed standard stocking densities to improve economic efficiency, resulting in complex coop environments—dim lighting, uneven floors, and obstacles such as feeders, waterers, sand, and feces.

The objective of on-farm data collection was to acquire images of floor-raised chickens exhibiting different behaviors for the subsequent training of a behavior recognition model. To ensure moderate lighting conditions, image collection was conducted between 10 a.m. and 3 p.m. from December 2022 to March 2024 using a Logitech C930c high-definition camera. During collection, the camera's shooting angle was kept at approximately 90 degrees, with manual adjustments to handheld and shooting angles to avoid light interference and cluttered backgrounds that could affect image quality. The shooting height and distance were controlled within 90–130 cm and 1–3 m, respectively. A schematic diagram is shown

in Figure 1. Real-time adjustments to the shooting angle and distance were made based on the captured footage to minimize occlusion and excessive distance from the chickens, ensuring high-quality behavioral images and a reliable foundation for constructing the behavior recognition model.

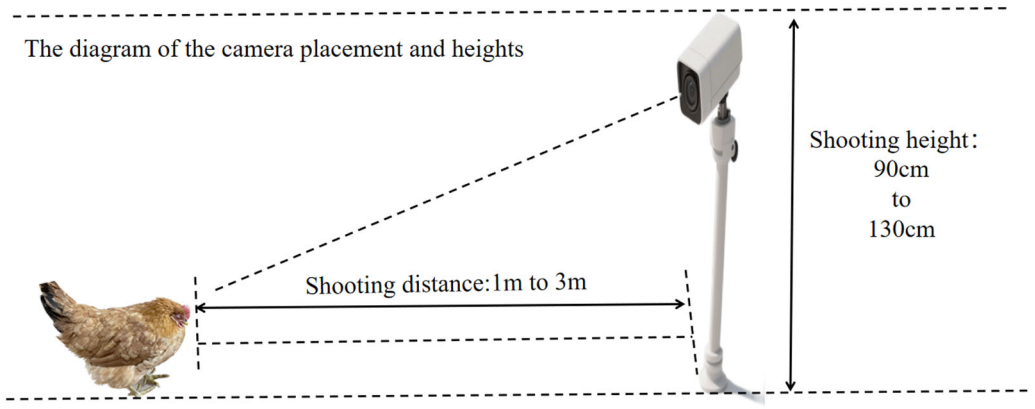

Figure 1. Diagram of the camera placement and heights.

Daily poultry behaviors, such as feeding and standing, are frequently observed and studied. By consulting literature on poultry behavioral characteristics, poultry diseases, and abnormal animal behaviors [31,32], and combining methods for identifying the health status and abnormal behaviors of yellow-feathered chickens, a standardized classification scheme for their behaviors was developed. As shown in Figure 2, the scheme categorizes behaviors into five types: Pecking, Resting, Walking, Dead, and Inactive. Inactivity is associated with sub-health conditions, such as leg disorders and contact dermatitis on the feet and breasts, which are related to inactivity in chickens [33].



Figure 2. Definitions of yellow-feathered chicken behaviors.

The definitions of the daily behaviors of chickens are shown in Table 1.

Table 1. Definitions of yellow-feathered chicken behaviors.

Behavior	Definition	Example
Pecking	Rapid, repetitive pecking motion using the beak to obtain food or explore surfaces	Figure 2
Resting	Stationary state for energy recovery, often accompanied by closed eyes or low activity	Figure 2b
Walking	Locomotion through coordinated limb movements	Figure 2c
Dead	Complete loss of vital signs (respiration, heartbeat, and neural reflexes)	Figure 2d
Inactive	Awake state with sustained lack of goal-directed activity, potentially indicating suboptimal health	Figure 2e

In this study, the annotation tool Labellmg [34] was used to annotate these five behaviors in each image using bounding box annotation. This tool allows users to draw rectangular boxes on images and label categories via a graphical interface. These operations are converted into coordinate and category information, saved as annotation files in formats such as PASCAL VOC or YOLO, and provide structured data input for training deep learning models. The core process includes image loading, interactive annotation, and data storage, supporting keyboard shortcuts and batch processing. The underlying implementation is based on Python 3.10 and PyQt6 for cross-platform interaction. To ensure the accuracy of the above dataset annotations, agricultural experts supervise and adjust the annotations during the annotation process. After annotation, we organized three rounds of checks by agricultural experts and students to further ensure the accuracy of the original annotations. After the annotation process, the annotation and category information for each image were exported to YOLO format annotation files. As shown in the figure, the numbers 0 to 4 correspond to the behaviors of Pecking, Resting, Walking, Dead, and Inactive, respectively. The training process requires a large amount of high-quality datasets, so it is necessary to perform data augmentation on the dataset used in this experiment. This is performed by processing the experimental images through methods such as translation transformation and adding noise. After data augmentation processing, a total of 1565 images were obtained, of which 1230 were original images. The annotated dataset, named ChickenData, was divided into a training set, validation set, and test set, containing 1126, 126, and 313 images, respectively.

2.2. Dual-Backbone Heterogeneous YOLOv11

The target detection system in floor-raised chicken farming faces two key technical challenges:

(1) Complex background interference and coexisting multi-scale target individuals significantly impact detection performance. Targets exhibit pronounced multi-scale characteristics due to varying distances from the camera, creating spatial scale heterogeneity that makes traditional detection algorithms struggle to maintain detection consistency across spatial domains.

(2) Cluttered farming environment noise conflicts with real-time processing requirements at the device end, especially under limited edge computing resources—the detection accuracy of existing models is significantly degraded by environmental interference.

To address these challenges, this study innovatively proposes the DualHet-YOLO, which is a deep optimization based on the YOLOv11 framework. The architecture incorporates a heterogeneous feature fusion mechanism and multi-scale attention modules specifically tailored to the characteristics of floor-raised chicken farming scenarios.

As shown in Figure 3, the DualHet-YOLO behavior detection model for floor-raised chickens employs a four-stage collaborative architecture. The Front-end Deep Feature Extraction Network generates multi-resolution feature maps through four-level residual modules: 80 × 80 × 256 (shallow details), 40 × 40 × 512 (mid-level semantics), and 20 × 20 × 1024 (high-level abstraction). It achieves resolution dimensionality reduction via strided convolutions and enhances feature representation capability through a channel expansion strategy, completing the progressive mapping of raw images to a high-dimensional semantic space. The Back-end Deep Feature Enhancement Network introduces a CBLinear channel reweighting mechanism, using the CBFuse module to perform cross-layer concatenation and upsampling operations on the multi-level features output by the front-end, constructing a composite feature pyramid that fuses spatial details and deep semantics while integrating visual features from multi-scale receptive fields through a parameterized fusion strategy.

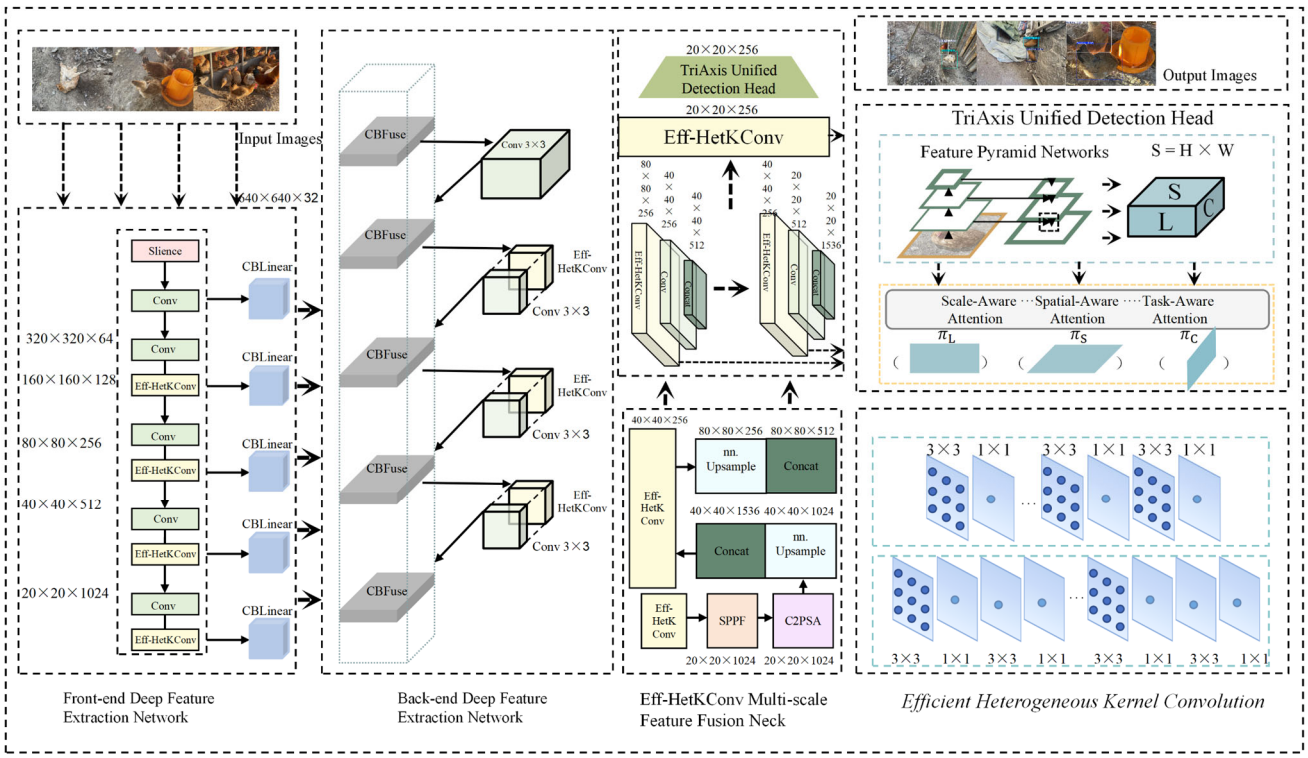

Figure 3. The structure of the DualHet-YOLO network.

The Eff-HetKConv Multi-scale Feature Fusion Neck adopts a bidirectional feature interaction mechanism with top-down semantic propagation and bottom-up detail supplementation strategies, enabling complementary enhancement of spatial-channel dual perception by integrating low-level edge/texture information with high-level target semantic features.

The TriAxis Unified Detection Head innovatively fuses three attention mechanisms—scale-aware, spatial-aware, and task-aware—adapting to target size variations through dynamic channel weighting, capturing pose diversity using deformable convolutions and achieving collaborative optimization of classification and localization through feature decoupling to realize cross-layer synchronous detection and precise localization of multi-scale, multi-morphology targets in free-range scenarios. This architecture innovatively integrates a trinity collaborative optimization framework of feedforward feature extraction, feedback feature enhancement, and multi-dimensional attention modeling. Under extreme scenario factors such as complex lighting changes, target occlusion, and

dense group distribution, it achieves precise capture of chicken posture transformations, movement trajectories, and group interaction behaviors through cross-layer feature progressive optimization and dynamic weight adaptation mechanisms, ultimately leading to a significant breakthrough in detection accuracy. The structural details and innovations of each module are described in detail below.

2.3. Dual-Path Feature Map Extraction Architecture

In the YOLOv11 object detection model, we innovatively introduce the overview of dual-path feature map extraction architecture to enhance the model's capability for feature information extraction and utilization. This architecture is inspired by an in-depth understanding of the information bottleneck problem in deep learning, as well as the ingenious application of invertible functions and auxiliary supervision mechanisms. Specifically, the dual-path feature map extraction architecture processes input data through two parallel backbone networks, each responsible for extracting feature information at different levels and semantics.

In practical design, the dual-path feature map extraction architecture is implemented through a series of carefully designed modules. The model begins with an initial convolutional layer to perform preliminary feature extraction and downsampling on the input image. Subsequently, the feature map is divided into two paths, entering two backbone networks, respectively. Each backbone network consists of multiple convolutional layers, ELAN-style modules (e.g., Eff-HetKConv), and feature fusion layers (CBFuse). These modules collaborate to ensure that each backbone network can deeply mine features from different perspectives. The following describes the hierarchical construction steps of the four stages of the dual-path feature map extraction architecture.

The first stage centers on the Front-end Deep Feature Extraction Network, constructing a basic feature extractor with four-level (P1–P5) downsampling using Eff-HetKConv modules to optimize computational efficiency via heterogeneous convolution. The main branch employs an improved Eff-HetKConv module, replacing standard convolutions with heterogeneous convolutions (HetConv). By mixing convolution kernels of different sizes (3×3 and 1×1), this approach enhances feature diversity while preserving more original information in cross-stage connections. The mathematical expression is shown as Equation (1), where F_{out} denotes the output features of the first stage and F_{in} represents the input features:

$$F_{out} = \text{Concat}(F_{in}, \text{HetConv}(F_{in})) \tag{1}$$

where the decomposition of the HetConv operation is shown in Equation (2), where $x \in \mathbb{R}^{B \times C \times H \times W}$ denotes the input feature map, and $\text{Conv}_{3\times3}$ and $\text{Conv}_{1\times1}(x)$ denote the 3×3 and 1×1 convolution operations:

$$\text{HetConv}(x) = \text{Conv}_{3\times3}(x) \oplus \text{Conv}_{1\times1}(x) \tag{2}$$

As shown in Figure 4, the second stage deploys a cross-branch routing mechanism within the Back-end Deep Feature Extraction Network using the CBLinear module to extract multi-scale features from each level of the front-end network and establish information pathways. The CBLinear module constructs a feature routing mechanism for the auxiliary branch, extracting features from the P5 layer of the main branch and performing linear projection on 5D features. This multi-scale feature aggregation strategy significantly enhances the information capacity of the auxiliary branch, providing a rich gradient signal source for subsequent fusion. The mathematical formulation of the second stage is expressed by Equation (3), where $x_i \in \mathbb{R}^{B \times C_i \times H \times W}$ denotes the input features of the i-th

layer, $W_i \in \mathbb{R}^{C_{out} \times C_i \times 1 \times 1}$ represents the parameters of the 1×1 convolution kernel, and N signifies the number of routing layers:

$$\text{CBLinear}(x) = \sum_{i=1}^{N} W_i \cdot x_i \tag{3}$$

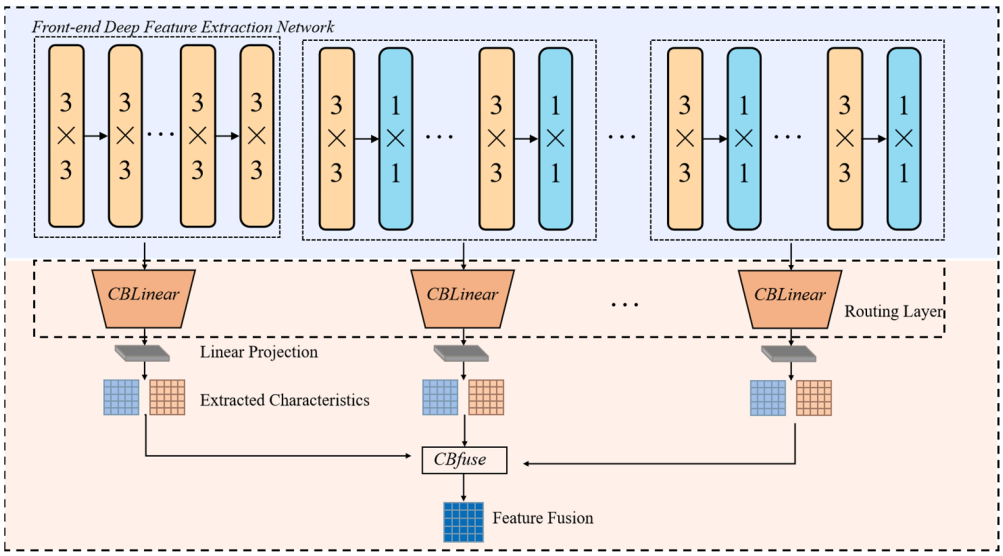

Figure 4. Cross-branch routing pathways with multi-scale feature aggregation.

In the third stage, the CBFuse module is used to achieve spatially adaptive fusion of features from the two networks, and a channel attention weighting strategy is adopted to enhance the integrity of gradient information. The CBFuse module employs a channel attention mechanism to realize the adaptive fusion of features. The specific implementation process is as follows: First, the current feature F_{main} of the main branch and the features $\{F^i_{aux}\}$ of the auxiliary branch are concatenated along the channel dimension. Then, the channel statistics are generated through global average pooling. Two fully connected layers are used to generate the channel weight vectors α and $\{\beta_i\}$, and weighted fusion is performed: $F_{fused} = \alpha F_{main} + \sum \beta_i F^i_{aux}$. This process can be formalized as Equation (4):

$$F_{fused} = \text{Sigmoid}(\text{FC}(\text{GAP}(F_{cat}))) \odot F_{cat} \tag{4}$$

where F_{cat} is the spliced feature and $\odot$ denotes the channel level multiplication.

In the fourth stage, heterogeneous convolution stacking and channel recalibration are performed on the basis of the fused features. Finally, a highly discriminative feature pyramid containing P3–P5 is output, completing the in-depth refinement and optimization of multi-scale object detection features. The network refines the fused features by stacking Eff-HetKConv modules. These modules utilize the gradient regularization signals provided by the auxiliary branch, effectively suppressing the feature degradation phenomenon. Their effect can be verified by the change in information entropy, which can be represented by Equation (5), where H represents the information entropy and $p(x)$ represents the feature probability distribution of the channel.

$$H\left(F_{final}\right) = -\sum p(x) \log p(x) \tag{5}$$

To enable information interaction and fusion within the overview of dual-path feature map extraction architecture and facilitate the construction of the four-stage network, we introduce routing layers (CBLinear) and feature fusion layers (CBFuse) at specific network

levels. Routing layers are responsible for linearly combining or routing feature maps from different levels, providing multi-scale feature inputs for subsequent feature fusion. Feature fusion layers, in turn, deeply integrate features from the two paths to ensure the main branch receives complete semantic information, avoiding information loss and unreliable gradient propagation. Based on this, this section presents the following two innovations.

2.3.1. Reversible Auxiliary Branching via Modular Realization of CBLinear

As shown in the code snippet of Table 2, the cross-branch linear transformations (CBLinear) module constructs five levels of feature routing channels, and each CBLinear layer contains the following core operations.

Table 2. Core operations of each CBLinear layer.

Algorithm format CBLinear Forward Propagation Require: Input feature map x ∈ R^{B × C_{in} × H × W}, in_channels, out_groups Ensure: Split feature maps {Gi} where Gi ∈ R^{B × g_i × H × W}, ∑g_i = out_groups
1: Initialize Conv1 × 1 layer ▷ Step1: Create a 1 × 1 convolutional kernel 2: Store groups configuration: G = [[g_1], [g_2],..., [g_n]] ▷ Preset grouping modes 3: Forward computation: ▷ Applying a 1 × 1 convolution ▷ Channel Dimension Segmentation 4: return OrderedDict{G_1: ..., G_n: ...} ▷ Maintaining group sequentiality

Let the output feature of the kth layer of the main branch be $F^k_{main} \in \mathbb{R}^{C_k \times H \times W}$, and the projection process of CBLinear can be formalized as Equation (6):

$$F^l_{aux} = \mathcal{P}_l\left(F^k_{main}\right) = \sum_{c=1}^{C_k} W_{l,c} * F^k_{main}[c] + b_l \tag{6}$$

where $W \in \mathbb{R}^{C_{out} \times C_{in} \times 1 \times 1}$ is the 1 × 1 convolution kernel parameter and $*$ denotes the channel-by-channel convolution operation. In particular, [9, 1, CBLinear, [[64, 128, 256, 512, 1024]]] denotes the extraction of 5 groups of features from the 9th layer of the main branch (P5 output), each with 64, 128, 256, 512, and 1024 channels, respectively. This multi-scale compression strategy can be explained by Equation (7):

$$\mathcal{P}(F) = \bigoplus_{i=1}^{5} \mathcal{T}_i(F), \mathcal{T}_i \in \mathbb{R}^{C_i \times C_{in}} \tag{7}$$

where $\oplus$ denotes channel dimensional splicing and Ti is a low-rank projection matrix whose parameters are learned through end-to-end training.

CBLinear assumes the role of a gradient distributor in backpropagation. Let the total loss function be L. The auxiliary branch gradient can be computed as Equation (8):

$$\frac{\partial \mathcal{L}}{\partial F^k_{main}} = \sum_{l=1}^{L} \frac{\partial \mathcal{L}}{\partial F^l_{aux}} \cdot \frac{\partial F^l_{aux}}{\partial F^k_{main}} \tag{8}$$

where L is the number of connected auxiliary branches. This multi-path gradient back propagation mechanism effectively mitigates the gradient vanishing problem.

2.3.2. Dynamic Weighting and Spatial Attention Mechanisms for CBFuse Modules

The dynamic weight generation process of the CBFuse module is shown in Figure 5. Given the main branch feature $F_{main} \in \mathbb{R}^{C\times H\times W}$ and N auxiliary features $\{F^i_{aux}\}^N_{i=1}$, the fusion process is divided into three steps: feature splicing, channel statistics generation, and dynamic weight calculation:

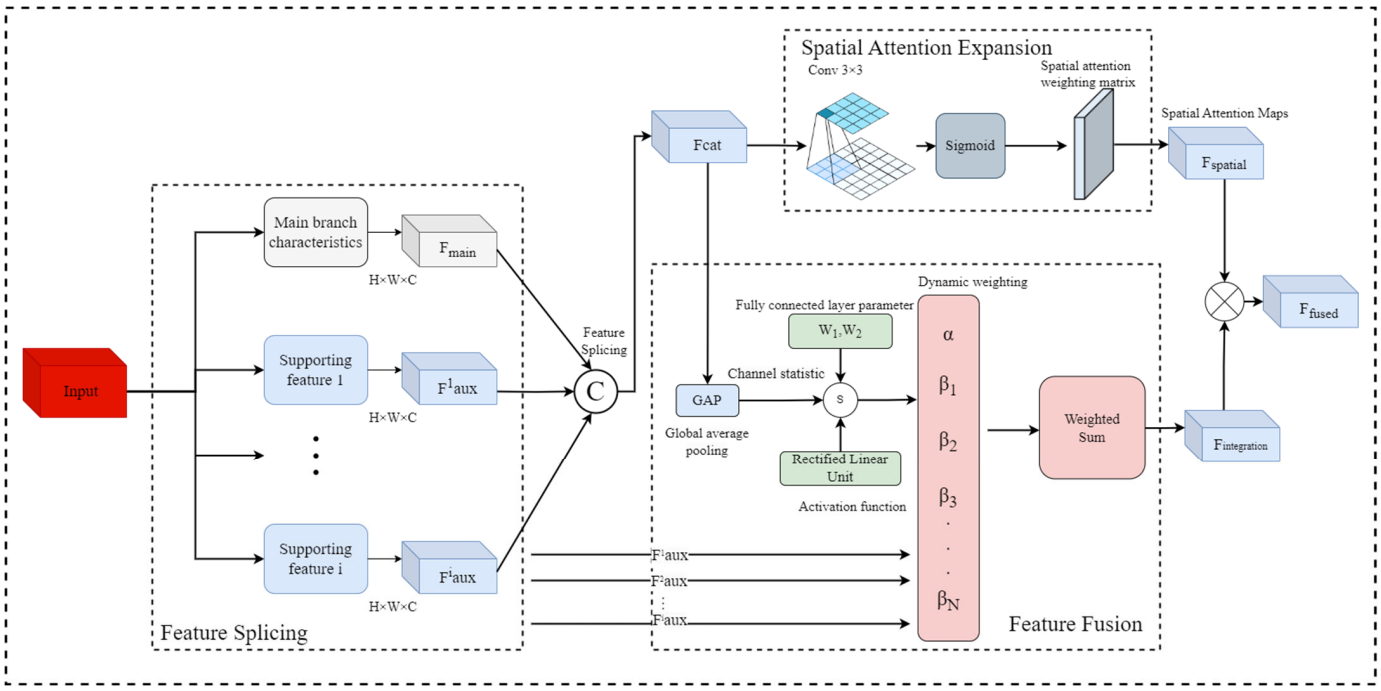

Figure 5. Generation of dynamic weights and introduction of spatial attention mechanisms.

Step 1: Feature splicing is described as Equation (9).

$$F_{cat} = \text{Concat}\left(F_{main}, F^1_{aux}, \ldots, F^N_{aux}\right) \in \mathbb{R}^{(C+\sum C_i)\times H\times W} \tag{9}$$

Step 2: Channel statistics generation is described as Equation (10).

$$s = \text{GAP}(F_{cat}) \in \mathbb{R}^{C+\sum C_i} \tag{10}$$

Step 3: Dynamic weight calculation is described as Equation (11).

$$[\alpha, \beta_1, \ldots, \beta_N] = \sigma(W_2 \cdot \delta(W_1 \cdot s)) \tag{11}$$

where $W_1 \in \mathbb{R}^{(C+\sum C_i)\times r}$ and $W_2 \in \mathbb{R}^{r\times(N+1)}$ are fully connected layer parameters, δ is the ReLU activation, and σ is the Sigmoid function. The compression ratio r is set to 16 to balance the computational complexity. In the specific implementation, CBFuse uses grouped convolution to improve efficiency, as shown in the code snippet of Table 3.

To further improve the performance of small target detection, we introduce a spatial attention mechanism in high-level CBFuse; $A_{spatial} \in \mathbb{R}^{B\times 1\times H\times W}$ denotes the spatial attention map as Equation (12):

$$A_{spatial} = \text{Sigmoid}(\text{Conv}_{3\times 3}(F_{cat}))F_{fused} = A_{spatial} \odot \left(\alpha F_{main} + \sum \beta_i F^i_{aux}\right) \tag{12}$$

Table 3. Grouped convolution for CBFuse.

Algorithm format CBFuse Forward Propagation
1: fused_map ← concat(X, dim = 1) [Step 1: Concatenate all input feature maps along the channel dimension] 2: conv_layer ← Conv2d(in_channels = sum(groups) out_channels=sum(groups), kernel_size = 1, groups = n) [Step 2: Define a grouped point-wise convolution layer] 3: fused_map ← conv_layer(fused_map) [Step 3: Apply the grouped point-wise convolution] 4: X′ ← split(fused_map, split_size = groups, dim = 1) [Step 4: Split the fused feature map back into the original number of groups] 5: return X′

2.4. YOLOv11 Efficient Heterogeneous Kernel Convolution

In the field of target detection, real time and accuracy of the model are required for floor chicken behavior recognition, and the C3K2 structure of the YOLOv11 model has limitations of computational efficiency and feature expression ability. In this paper, we propose the Eff-HetKConv structure, which integrates the advantages of heterogeneous convolutional kernels, reduces the computational complexity, and improves the efficiency of feature extraction so as to provide a highly efficient solution for the recognition of floor-raised chicken behaviors.

2.4.1. Deficiencies in the C3K2 Structure

In the floor chicken behavior recognition scenario, the C3K2 structure in the YOLOv11 model has obvious deficiencies that restrict the effect of practical application. The C3K2 structure has a bottleneck in computational efficiency. Despite the optimization of feature extraction, it still consumes more computational resources and time when dealing with large-scale data and complex scenarios, making it difficult to meet real-time requirements. In terms of the flexibility and diversity of feature extraction, the C3K2 structure mainly extracts features through two convolutional layers, and the fixed structure is difficult to adapt to the diversified feature requirements in different scenarios. It cannot fully capture the complex features of the diverse behaviors of floor chickens, which affects the accuracy of behavior recognition. In addition, the C3K2 structure is not satisfactory enough in terms of parameter utilization efficiency, and there is a parameter redundancy problem, which increases the storage and computation overhead of the model, limiting its wide application on resource-limited devices.

2.4.2. Eff-HetKConv Structure Principle

To address the issues in the C3K2 structure, we propose an improved Eff-HetKConv structure. The core principle of Eff-HetKConv is to construct convolutional layers using heterogeneous convolution kernels. Specifically, in a single convolutional layer, convolution kernels of different sizes, such as 3×3 and 1×1 kernels, are used simultaneously. By reasonably allocating the usage ratios of these two types of convolution kernels across channels, we can retain the ability of 3×3 convolution kernels to capture local spatial features and utilize 1×1 convolution kernels to reduce the computational cost while performing feature fusion and dimensionality reduction between channels.

Compared with the C3K2 structure, Eff-HetKConv not only has an advantage in computational efficiency but is also more powerful in feature extraction. The 3×3 convolution kernels are responsible for capturing local spatial correlations, while the 1×1 convolution kernels can recombine and weight features in the channel dimension, enabling the model to learn more discriminative feature representations. This combination allows Eff-HetKConv

to extract richer and more diverse features at different feature levels, which helps improve the model's ability to recognize target behaviors in complex scenarios. Figure 6 shows the difference between the standard filter and the Eff-HetKConv filter.

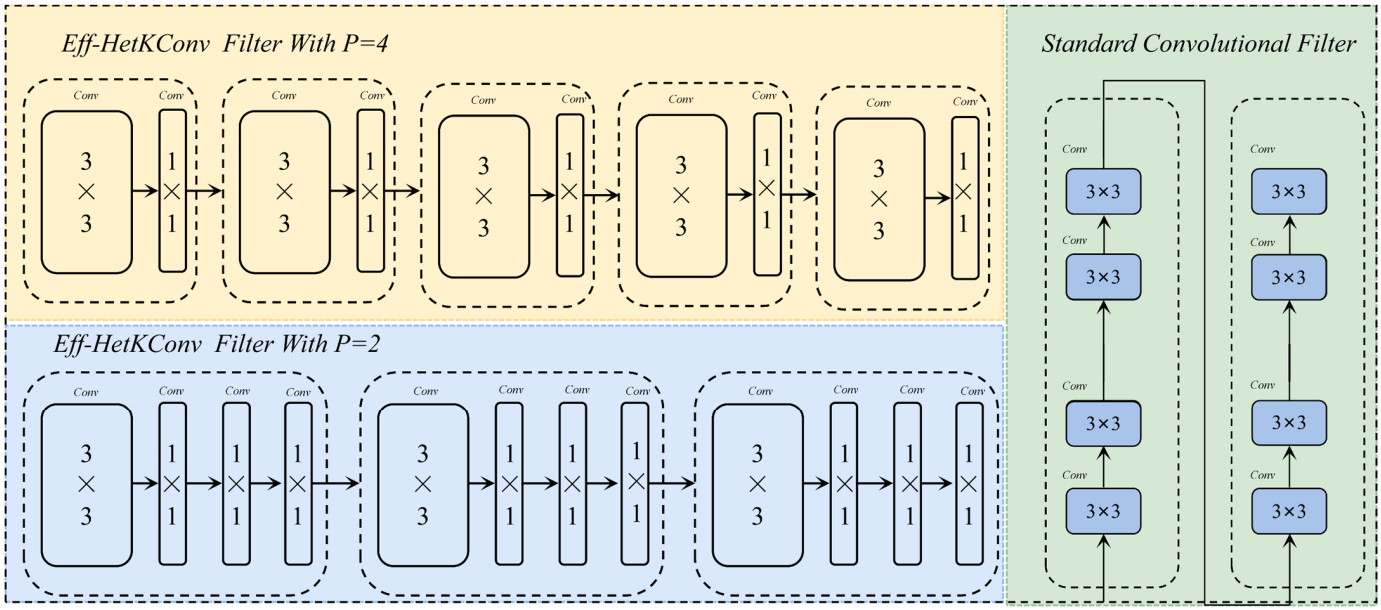

Figure 6. Difference between quasi-filters and Eff-HetKConv filters.

Eff-HetKConv achieves computational optimization by mixing convolutional kernels of different scales. For a convolutional layer with M number of input channels and N number of output channels, each output channel uses P-grouped heterogeneous kernels, 3×3 convolutional kernels for $1/P$ scaled channels, and 1×1 convolutional kernels for (1-1/P)-scaled channels. The FLOPs equation is shown as Equation (13):

$$L_{Eff} = \underbrace{\frac{D_o^2 \cdot M \cdot N \cdot 9}{P}}_{3\times3\text{section}} + \underbrace{D_o^2 \cdot N \cdot \left(M - \frac{M}{P}\right)}_{1\times1\text{section}} \tag{13}$$

where D_o is the output feature map size and 9 corresponds to the number of parameters in the 3×3 kernel. When compared to the computational complexity of the traditional C3K2 architecture, the scenario involving two 3×3 convolutional layers is described by Equation (14).

$$FL_{C3K2} = D_o^2 \cdot M \cdot N \cdot 9 \cdot 2 \tag{14}$$

The speed improvement ratio is shown in Equation (15).

$$Speedup = \frac{FL_{C3K2}}{FL_{Eff}} = \frac{18}{\frac{9}{P} + \left(1 - \frac{1}{P}\right)} \tag{15}$$

In the floor chicken behavior recognition task, the Eff-HetKConv structure exhibits multiple advantages that make it a more suitable convolutional structure for this task.

The Eff-HetKConv structure demonstrates significant advantages in the task of floor-raised chicken behavior recognition. By employing heterogeneous convolution kernels and reducing the usage ratio of large kernels, this structure substantially decreases the computational load and inference time, meeting the demand for real-time processing of massive data and providing timely and effective support for breeding management. In terms of feature extraction, Eff-HetKConv integrates the advantages of different-sized convolution

kernels, enabling it to capture local spatial features while obtaining more representative feature representations through channel fusion, thereby improving the accuracy of behavior recognition. For example, during chickens' foraging behavior, 3×3 kernels capture local action features, while 1×1 kernels integrate channel features to highlight behavior-relevant dimensions. Additionally, Eff-HetKConv reduces parameter redundancy by rationally allocating the use of convolution kernels across channels, making the model more compact and facilitating deployment on resource-constrained devices—thereby lowering hardware costs and deployment complexity. Overall, through optimizing computational efficiency, enhancing feature extraction capability, and improving model compactness, Eff-HetKConv provides an efficient, accurate, and practical solution for floor-raised chicken behavior recognition.

2.4.3. Co-Optimization of Two-Way Feature Map Extraction Architecture with Eff-HetKConv

Deformable Convolution is introduced before the CBLinear module to spatially align the feature maps of the auxiliary branch with the spatial location of the main branch. Since the heterogeneous convolution kernels (3×3 and 1×1) of Eff-HetKConv may lead to inconsistencies in the receptive fields, spatial alignment is a critical step to ensure the accuracy of feature fusion. This problem can be effectively solved by dynamically adjusting the spatial distribution of the feature map through deformable convolution. F_{aligned} denotes the aligned feature map, F_{aux} denotes the feature map of the auxiliary branch, and DeformConv denotes the deformable convolution operation. The relationship between A and B is described as Equation (16).

$$F_{\text{aligned}} = \text{DeformConv}(F_{\text{aux}}) \tag{16}$$

To balance the computational efficiency and feature expression ability, a computationally aware weight attenuation factor is introduced. This factor changes how much weight is reduced based on how complex Eff-HetKConv and C3K2 are, helping to prevent the model from focusing too much on the part that requires more computation during optimization. λ denotes the weight attenuation factor, and $\text{FLOPs}_{\text{Eff-HetKConv}}$ and $\text{FLOPs}_{\text{C3K2}}$ denote the Eff-HetKConv and C3K2, respectively, in terms of floating-point operations. The calculation method for λ is described as Equation (17).

$$\lambda = 0.01 \cdot \frac{\text{FLOPs}_{\text{Eff-HetKConv}}}{\text{FLOPs}_{\text{C3K2}}} \tag{17}$$

Since the heterogeneous convolutional kernel of Eff-HetKConv may lead to large differences in the gradients of different branches, a two-branch gradient normalization strategy is designed. Normalizing the gradient magnitude ensures that the contributions of the main and auxiliary branches to the loss function are balanced during the training process, which improves the stability of training and convergence speed.

With the above adaptation strategy, Eff-HetKConv and the dual-backbone feature extraction architecture achieve a synergistic optimization, which retains the advantages of heterogeneous convolutional kernels in feature extraction and ensures the computational efficiency and training stability of the model. These improvements provide a more efficient and reliable solution for target detection tasks in complex scenes.

2.5. TriAxis Unified Detection Head

The initial detection head of the YOLOv11 model shows remarkable efficiency and effectiveness but has certain limitations in floor-raised chicken behavior recognition. These limitations mainly appear in its inadequate handling of objects at different scales, limited

ability to capture spatial relationships, and inflexible adaptation to specific tasks. These issues affect the model's detection accuracy and robustness in complex situations of floor-raised chicken behavior. To solve these problems, we put forward a new object detection head: the TriAxis Unified Detection Head. It combines three attention mechanisms—scale-aware, spatial-aware, and task-aware—into one detection head. This design helps the model better deal with the complexities of floor-raised chicken behavior recognition. The structure of the TriAxis Unified Detection Head is described below.

The TriAxis Unified Detection Head builds a unified framework by combining three attention mechanisms: scale-aware, spatial-aware, and task-aware. These mechanisms are applied in sequence to the feature tensor to strengthen its representation, thus enhancing the accuracy of target detection.

Given a feature tensor $F_{in} \in \mathbb{R}^{L_{level} \times S_{space} \times C_{channel}}$, where L_level stands for the number of feature levels, S_{space} stands for the spatial dimensionality (height × width), and C_channel stands for the number of channels, the TriAxis Unified Detection Head applies the following three attention functions consecutively. The relationship between the three attention functions is shown in Equation (18):

$$W_{out}(F_{in}) = \pi_{C_{channel}}\left(\pi_{S_{space}}\left(\pi_{L_{level}}(F_{in}) \cdot F_{in}\right) \cdot F_{in}\right) \tag{18}$$

(1) Scale-aware attention ($\pi_{L_{level}}$): focuses on the feature level dimension, dynamically fuses features at different scales, and adjusts the weights of features at each level based on semantic importance.
(2) Space-aware attention ($\pi_{S_{space}}$): applied to the spatial dimension, learns discriminative representations of different spatial locations, helps the model to focus on the relevant regions, and better captures geometric transformations and spatial configurations of objects.
(3) Task-aware attention ($\pi_{C_{channel}}$): operating on the channel dimension, directs various feature pathways to prioritize distinct tasks. This allows the model to adjust resource distribution based on input, catering to diverse detection requirements like categorization, boundary box localization, and key point identification.

2.5.1. The Triple Attention Mechanism of Scale, Space, and Tasks

The equation for the scale-aware attention module is shown as Equation (19):

$$\pi_{Level}(Fin) \cdot Fin = \sigma\left(f\left(\frac{1}{S_{space,C_{channel}}} \sum_{space,C_{channel}} Fin\right)\right) \cdot Fin \tag{19}$$

where f (x) is modeled using a 1 × 1 convolutional layer as a linear function and $\sigma(x) = max(0, min(1, x + 1/2))$ serves as a hard sigmoid function. This module learns the relative importance of different semantic layers to enhance the feature representation of objects at different scales.

The space-oriented perception attention module is split into two phases as Equation (20):

$$\pi_{S_{space}}(Fin) \cdot Fin = \frac{1}{L_{\text{level}}} \sum_{l=1}^{L_{\text{level}}} \sum_{k=1}^{K} w_{l,k} \cdot Fin(l; p_k + \Delta p_k; c) \cdot \Delta m_k \tag{20}$$

where K represents the count of sparsely sampled locations, $p_k + \Delta p_k; c$ denotes the location modified by the self-learned spatial offset Δp_k, and Δm_k signifies the self-learned importance weight at the position p_k. This module concentrates on distinctive areas and flexibly combines features from various layers at the same spatial position.

The equation for the task-aware attention module is shown as Equation (21):

$$\pi_{C_{channel}}(Fin) \cdot Fin = max(\alpha_1(Fin) \cdot Fin_c + \beta_1(Fin), \alpha_2(Fin) \cdot Fin_c + \beta_2(Fin)) \quad (21)$$

where Fin_c is the feature slice of the cth channel, and $[\alpha_1, \alpha_2, \beta_1, \beta_2]^T = \theta(\cdot)$ is a hyperfunction for learning to control the activation threshold. The module dynamically switches the feature channels to adapt to different tasks, improving the model's ability to adapt to different detection demands.

2.5.2. Full-Dimensional Dynamic Triple Focus Module

As shown in Figure 6, the OmniDyna TriFocus Block, the core component in the TriAxis Unified Detection Head, is a composite structure based on multi-dimensional attentional synergies, whose core design revolves around the 3D tensor output from the feature pyramid, which is finely tuned dynamically in terms of hierarchical, spatial, and channel dimensions, respectively.

In the scale-aware attention module, the system first compresses the information in the spatial and channel dimensions through global average pooling to generate a feature vector representing the importance of each hierarchical level. Subsequently, a lightweight 1×1 convolutional layer is used to learn the correlation weights between hierarchical levels, and a hard Sigmoid function is applied to constrain the weights to the range of (0, 1). This process enables the model to dynamically allocate the contribution of feature levels according to the target size. For example, it enhances the representation ability of shallow features for small targets while suppressing the redundant responses of deep features for large targets.

The spatial-aware attention module achieves dynamic receptive field adjustment through a deformable convolution mechanism. Based on the standard 3×3 convolution kernel, this module predicts the offset parameters (Δp) using the middle-layer features, guiding the sampling points of the convolution kernel to adaptively shift towards the key deformation regions of the target (such as vehicle tires and animal limbs). Meanwhile, it assigns different weights to the sampling points by combining them with the mask parameters (Δm). In particular, after calculating the independent offset for features at different levels, the module forms the final spatial attention map through cross-layer weighted aggregation, effectively capturing the spatial context associations of multi-scale targets.

The final task-aware attention module constructs a non-linear mapping relationship in the channel dimension through a fully connected layer. First, it generates two sets of learnable affine transformation parameters (α, β). Then, it performs linear transformation and maximum value fusion operations on the channel features. This design allows the module to dynamically select and enhance key channel features according to different task requirements, such as classification and regression. For example, it enhances the responses of channels with strong semantic discrimination while suppressing the interference of noise channels.

Figure 7 illustrates the integrated application of the OmniDyna TriFocus Block in a single-stage object detector through a clear modular design. The upper region shows the cascaded arrangement of multiple focus modules (e.g., TL, TS, TC), with arrows annotating the feature flow direction to form a serial processing pipeline for the "scale-aware–spatial-aware–task-aware" attention mechanisms. Within each head block, simplified symbols (e.g., π) represent the attention computation process, while lateral connections between hierarchical levels enable multi-stage feature optimization. The overall architecture presents a progressive enhancement path from basic feature input to multi-task output, demonstrating a systematic integration of attention mechanisms for hierarchical feature refinement.

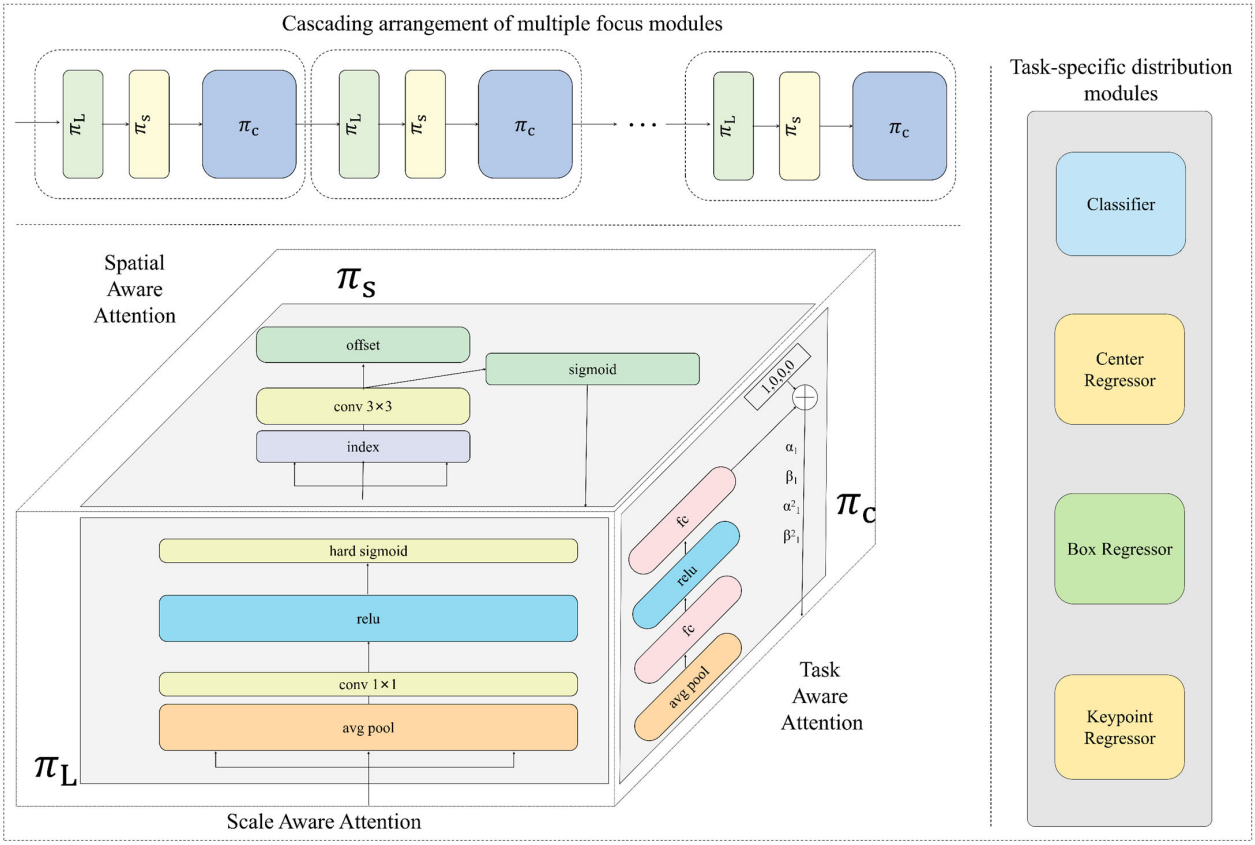

Figure 7. Generation of dynamic weights and introduction of spatial attention mechanisms. The architecture of the OmniDyna TriFocus Block.

The right region corresponds to the specific task distribution modules of the focus block, including the Classifier, Center Regressor, Box Regressor, and Keypoint Regressor. The figure uses a vertically aligned layout to visually present the mapping between outputs of different head blocks and task modules: for example, the scale-aware head block labeled "TL" primarily serves the box regression task, while the task-aware head block "TC" preferentially associates with the Classifier. This design reflects the directional adaptation characteristics of different attention mechanisms to detection tasks while retaining the efficiency of feature sharing.

From a data flow perspective, basic features first undergo cascaded processing by the left-side focus modules, sequentially completing multi-scale feature fusion, spatial deformation modeling, and channel semantic filtering. The optimized features are then distributed to the right-side task modules through a branching structure. For instance, the Keypoint Regressor receives feature inputs from both the spatial-aware (TS) and task-aware (TC) head blocks, capturing both geometric offset information of target local regions and enhancing channel responses strongly correlated with keypoint semantics. The Classifier, in contrast, heavily relies on discriminative channel features refined by the task-aware head block.

Through dynamic computation of attention weights, different dynamic modules enable the detector to flexibly balance the needs for multi-scale target detection, dense spatial localization, and complex semantic understanding. This approach maintains the efficiency of single-stage detectors while significantly improving detection accuracy for deformable targets like floor-raised chickens through multi-dimensional attention mechanisms.

2.6. Proportional Scale IoU: Adaptive Scale Perceptual Loss Function for Behavior Recognition of Floor-Raised Chickens

In the task of daily behavior recognition for floor-raised chickens, object detection models must accurately capture multi-scale and multi-morphology target features of flocks in different behavioral states (e.g., walking, pecking, fighting). However, traditional IoU-based loss functions in YOLOv11 (such as CioU and SIoU) primarily focus on the geometric relationships between predicted and ground-truth boxes, failing to fully exploit the impact of inherent target attributes (e.g., aspect ratio, absolute scale) on the regression process. To address this issue, this section proposes the Proportional Scale IoU (Pro-Scale IoU) loss function, which significantly improves the behavior recognition accuracy of the lightweight model DualHet-YOLO in complex scenarios by introducing a target box's morphological proportional factor and scale-adaptive factor.

The core concept of Pro-Scale IoU stems from in-depth observations of behavior recognition scenarios: First, the behavioral patterns of floor-raised chickens exhibit significant morphological differences—for example, wing-spreading behavior presents a floor morphology with an aspect ratio > 1, while standing behavior shows an approximately square contour. Second, the target scales of different behaviors span a wide range (e.g., the area of a full-body detection box can be over six times that of a local action box). Traditional loss functions do not explicitly distinguish these characteristics during regression, leading to systematic biases in the localization of morphology-sensitive behaviors. Pro-Scale IoU solves this problem through a dual-path adaptive mechanism: the Proportional Factor dynamically adjusts coordinate regression weights based on the aspect ratio of ground-truth boxes, enabling the model to focus more on offset correction in the long-edge direction; the Scale Factor introduces non-linear scaling based on target absolute size, enhancing the regression sensitivity for small targets. After embedding the Pro-Scale IoU module into YOLOv11's regression head, it generates spatially adaptive loss surfaces by real-time parsing of the morphological and scale features of GT boxes, guiding the network to prioritize the optimization of localization errors in key dimensions.

The advantages of Pro-Scale IoU in floor-raised chicken behavior recognition are reflected in four aspects: First, it improves localization accuracy—by introducing proportional factors and geometric constraints, it provides effective gradients even when there is no overlap between the target and predicted boxes, alleviating the gradient vanishing issue of traditional IoU and reducing misdetections and omissions. Second, it enhances multi-scale adaptability—dynamically adjusting loss weights balances the detection performance for targets of different sizes (e.g., chicks vs. adult chickens), avoiding missed small targets and misaligned large targets. Third, it improves generalization ability—geometric feature modeling endows the model with robustness against complex scenarios such as lighting changes and occlusions, ensuring stable recognition in real farming environments. Fourth, it accelerates model convergence—the loss function, which integrates target scale and shape, provides a clearer optimization direction, shortening the training cycle compared to traditional methods and facilitating rapid iterative deployment. Through refined spatial relationship modeling, this loss function balances detection accuracy and efficiency, providing a reliable technical foundation for the automated management of floor-raised chicken behavior analysis.

The core innovation of Pro-Scale IoU lies in the establishment of a dual-domain resolution mechanism for morphological scale and target scale. The morphological scale factor calculation path is shown in Equation (22):

$$\kappa_p = \frac{2 \cdot \left(\phi^{gt}\right)^{\delta}}{\left(\phi^{gt}\right)^{\delta} + \left(\psi^{gt}\right)^{\delta}}, \quad \tau_p = \frac{2 \cdot \left(\psi^{gt}\right)^{\delta}}{\left(\phi^{gt}\right)^{\delta} + \left(\psi^{gt}\right)^{\delta}} \tag{22}$$

where ϕ^{gt} and ψ^{gt} denote the width and height of the real frame, respectively, and δ is the morphological sensitivity coefficient. When the target width-to-height ratio $r = \frac{\phi^{gt}}{\psi^{gt}} > 1$, $\kappa_p > \tau_p$, the model increases the coordinate error weight in the width direction (long side). The scale adaptive factorization path is shown in Equation (23), where, $A_{gt} = \phi^{gt} \cdot \psi^{gt}$ is the target area, A_{base} is the average area of the dataset, and ζ is the scale gain factor. This factor is able to produce a loss amplification effect for small targets.

$$\mu_s = 1 + \frac{\zeta}{\sqrt{A_{gt}/A_{base}}} \tag{23}$$

Reconstruct the Pro-Scale IoU loss function by embedding the above factors into the SloU framework. The reconstruction process is shown in Equation (24).

$$L_{PS-IoU} = 1 - IoU + \frac{\Delta_{prop} + \Omega_{scale}}{2} \tag{24}$$

The calculation method for the morphological correction term Δ_{prop} is described as Equation (25), where $\varepsilon_x = \left(\frac{x_c - x_c^{gt}}{\sigma^2}\right)^2$, $\varepsilon_y = \left(\frac{y_c - y_c^{gt}}{\psi^{gt}}\right)^2$, and ϕ^{gt} and ψ^{gt} are the width and height of the minimum enclosing frame. The design gives a higher penalty weight to coordinate deviations in the long side direction (e.g., the width of the spreading behavior).

$$\Delta_{prop} = \kappa_p \cdot \left(1 - e^{-\varepsilon_x}\right) + \tau_p \cdot \left(1 - e^{-\varepsilon_y}\right) \tag{25}$$

The calculation method for the scale correction item Ω_{scale} is shown as Equation (26), where $\nu_\phi = \frac{|\phi - \phi^{gt}|}{\max(\phi, \phi^{gt})}$, $\nu_\psi = \frac{|\psi - \psi^{gt}|}{\max(\psi, \psi^{gt})}$. When detecting small targets, μ_s significantly increases the contribution of the aspect error.

$$\Omega_{scale} = \mu_s \cdot \left[\left(1 - e^{-\nu_\phi}\right)^4 + \left(1 - e^{-\nu_\psi}\right)^4\right] \tag{26}$$

GT Feature Extraction Layer: Real-time parsing of each target's ϕ^{gt}, ψ^{gt}, and A_{gt} from annotated data, with parallel computation of shape ratio coefficients κ_p, τ_p, and scale factor μ_s.

Dynamic Weight Fusion Layer: Injecting κ_p into the coordinate regression branch to achieve per-anchor weight allocation via matrix broadcasting mechanism; simultaneously performing Hadamard product operation between μs and width–height loss terms to realize scale-aware enhancement.

Differentiable Loss Calculation Layer: Employing automatic differentiation technology to seamlessly integrate the composite loss term L_{PS-IoU} into the model's backward propagation pipeline, ensuring stable convergence during training.

In the backpropagation stage, the gradient computation of the Pro-Scale IoU exhibits spatial anisotropy properties. Taking the width direction gradient as an example, the width direction gradient is shown as Equation (27).

$$\frac{\partial \mathcal{L}}{\partial \phi} = \underbrace{\mu_s \cdot \frac{4\left(1 - e^{-\nu_\phi}\right)^3}{\phi + \phi^{gt}}}_{\text{Scale Enhancements}} + \underbrace{\frac{2\delta\left(\phi^{gt}\right)^{\delta-1}\left(\psi^{gt}\right)^{\delta}}{\left[\left(\phi^{gt}\right)^{\delta} + \left(\psi^{gt}\right)^{\delta}\right]^2}}_{\text{Morphologically sensitive terms}} \tag{27}$$

The equation shows that, when dealing with targets with large aspect ratios, the morphology-sensitive term causes the network to preferentially correct the width error, while the scale-enhancing term produces a larger gradient magnitude and accelerates the model convergence when encountering small-scale targets.

3. Results

3.1. Baseline Selection Basis

During the model selection process for target detection of floor-raised yellow-feathered chickens, we ultimately chose YOLOv11n as the baseline through a comparative analysis of the performance and efficiency of different models. This decision was grounded in a comprehensive evaluation of model accuracy, parameter count, and computational complexity. As illustrated in Table 4, YOLOv11n demonstrated excellent performance in metrics such as mAP@50 throughout the training process. Compared with the s, m, and l variants, YOLOv11n achieved relatively high accuracy at the early training stages and maintained stability in the later phases. Although the x variant exhibited a slightly higher mAP value than YOLOv11n, its parameter count was significantly larger—2.6 million for YOLOv11n versus as high as 56.8 million for the x variant. This disparity would lead the x variant to consume more memory and computational resources in practical applications, rendering it unsuitable for deployment and scenarios with strict real-time requirements. Additionally, YOLOv11n had a FLOPs count of 6.4 G, far lower than those of the s, m, l, and x variants, indicating that it can perform inference at a faster speed under the same hardware conditions, thus meeting the demands of real-time target detection. By integrating considerations of model accuracy, parameter count, and computational complexity, YOLOv11n strikes a favorable balance between performance and efficiency. Therefore, it was selected as the baseline, providing a robust foundation for subsequent research and optimization.

Table 4. Comparison of YOLOv11n, YOLOv11s, YOLOv11m, YOLOv11l, and YOLOv11x Models.

Method	$mAP_{0.5}$-Epochs50	$mAP_{0.5}$-Epochs100	$mAP_{0.5}$-Epochs150	$mAP_{0.5}$-Epochs200	Parameters (M) ↓	FLOPs (G) ↓
YOLOv11n	52.9	67.9	72.8	78.1	2.6	6.4
YOLOv11s	35.0	57.8	69.1	72.5	9.4	21.6
YOLOv11m	58.4	67.6	71.5	73.0	20.0	67.7
YOLOv11l	44.5	62.1	77.3	75.9	25.3	86.6
YOLOv11x	48.5	68.0	71.6	78.3	56.8	194.4

3.2. DualHet-YOLO Model Performance

3.2.1. Performance of the DualHet-YOLO Method

The performance comparison between the benchmark model YOLOv11n and the improved model DualHet-YOLO in terms of Precision (P), Recall (R), and mAP@50 metrics is demonstrated in Figure 8. As can be seen from the figure, the P and R values of the two models exhibit significant fluctuations at the beginning of training (about the first 70 cycles), but the fluctuations gradually decrease as the training progresses. After the 40th cycle of training, the DualHet-YOLO model starts to consistently outperform the baseline model, YOLOv11n. Eventually, the P-value and R-value of the DualHet-YOLO model reach 83.2% and 78.1%, respectively, which are significantly higher than the baseline model.

On the mAP@50 metric, the curves of the two models show rapid changes in the first 60 cycles of training and then gradually stabilize. The mAP@50 curve of the DualHet-YOLO model consistently outperforms that of the baseline model throughout the training process and reaches a stable value of 84.1% at the end. This result indicates that the DualHet-YOLO model has higher accuracy and stability in the target detection task.

As shown in Figure 9, the number of parameters of the DualHet-YOLO model is reduced by 14.6% and FLOPs by 29.2% after the addition of Eff-HetKConv. In conclusion,

the DualHet-YOLO model achieves further optimization, furthering the accuracy of the original model while retaining the advantages of the original model's light weight.

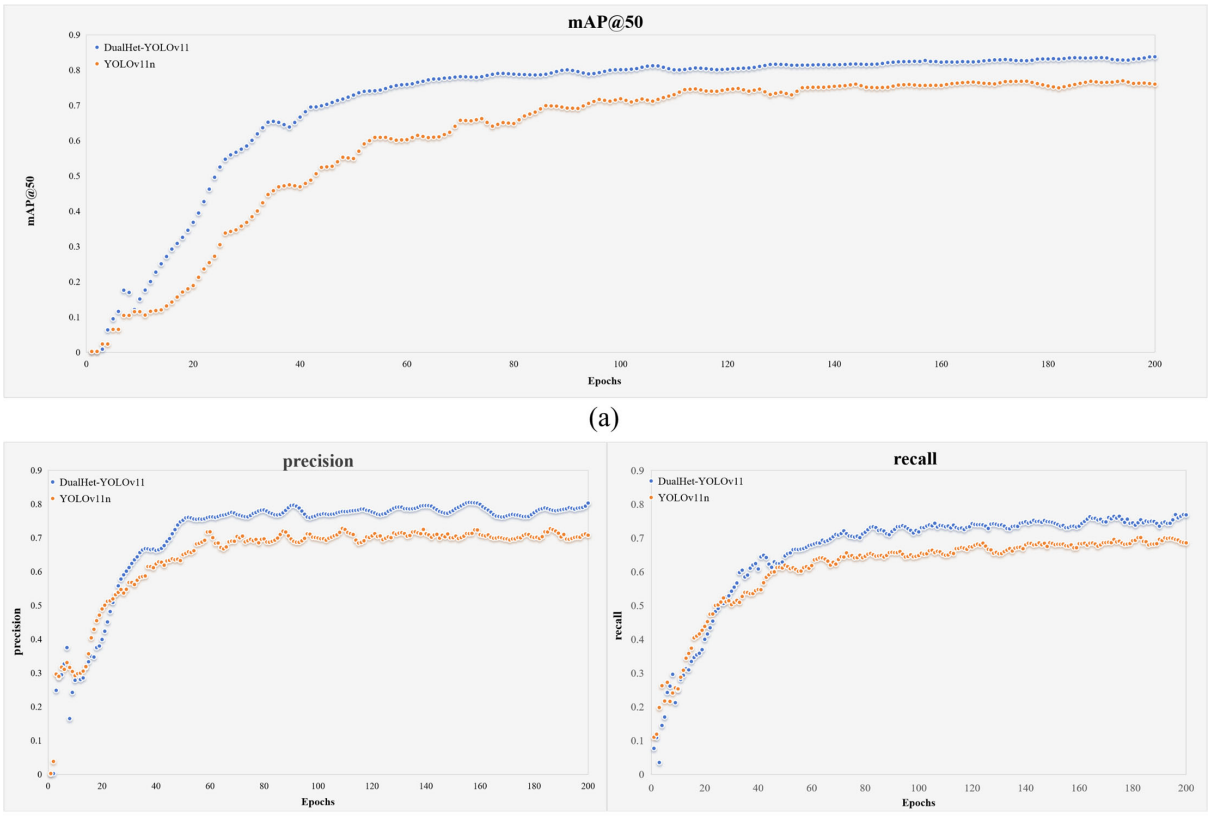

Figure 8. Performance comparison between the benchmark model YOLOv11n and the improved model DualHet-YOLO on Precision, Recall, and mAP@50 metrics: (**a**) mAP@50; (**b**) Recall; (**c**) Precision.

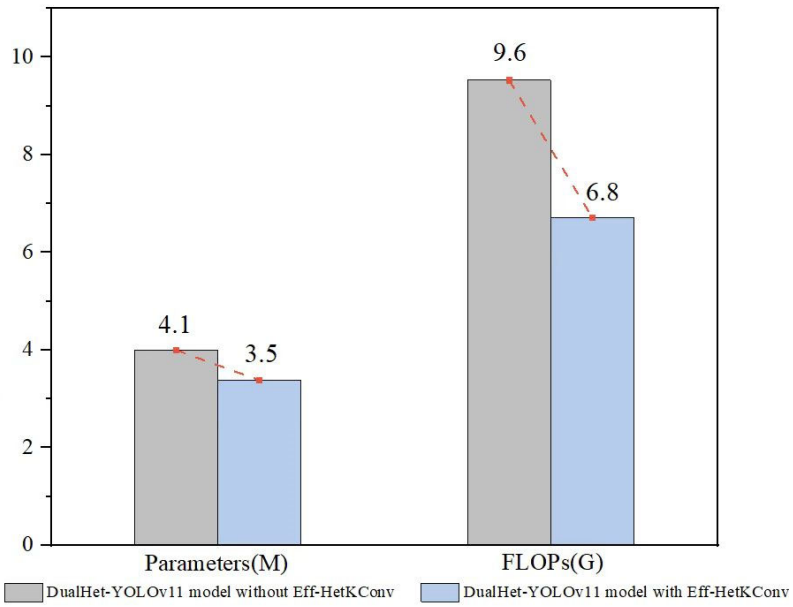

Figure 9. Comparison of model parameters before and after adding Eff-HetKConv to DualHet-YOLO (The red dotted line indicates a downward trend.).

3.2.2. Ablation Experiments on the Model's Performance

To thoroughly assess the role of each component in the DualHet-YOLO system regarding its overall effectiveness, we developed a set of component-wise experiments, as illustrated in Table 5. These trials confirm the usefulness and importance of each element by methodically incorporating or eliminating particular parts and examining their effects on system effectiveness. The experiments utilize the YOLOv11n reference model as a foundation, with stepwise introductions of the TriAxis-Head, Dual backbone, Eff-HetKConv, and Pro-ScaleIoU elements. The evaluation metrics encompass precision (P), recall (R), average precision at an IoU threshold of 0.5 (mAP@50), and average precision across IoU thresholds from 0.5 to 0.95 (mAP@50–95). Additionally, the model's parameter count and computational requirements (FLOPs) are recorded.

Table 5. Results of DualHet-YOLO ablation experiments in the floor chicken model.

Number	TriAxis-Head	Dual Backbone	Eff-HetKConv	Pro-Scale IoU	P	R	mAP0.5	mAP0.5–0.95	Parameters (M) ↓	FLOPs (G) ↓
1					72.8	71.2	78.5	52.1	2.6	6.4
2	√				74.2	70.4	80.0	55.5	2.3	7.20
3		√			78.9	74.5	82.3	57.4	4.2	10.8
4	√	√			80.1	74.6	83.4	57.7	4.1	9.60
5	√	√	√		83.2	76.1	83.8	57.9	3.5	6.8
6	√	√	√	√	83.4	77	84.1	58.2	3.5	6.8

Experimental results show that the TriAxis-Head significantly enhances the model's multi-task adaptability, increasing precision from 72.8% to 74.2% and mAP@50 from 78.5% to 80.0%. The Dual-path backbone further strengthens multi-scale object detection capabilities, boosting recall from 70.4% to 74.5% and improving mAP@50–95 from 52.1% to 57.4%. The Eff-HetKConv reduces computational complexity while maintaining efficient feature extraction, decreasing the parameter count from 4.2 M to 4.1 M and FLOPs from 10.8 G to 9.60 G, with precision increasing to 80.1%. The ProScaleIoU module significantly improves small-object detection accuracy, elevating precision to 83.2% and recall to 76.1%.

The full model (with all modules integrated) achieves optimal performance across all metrics, reaching 83.4% precision, 84.1% mAP@50, and 58.2% mAP@50–95, while maintaining low parameter count and computational complexity (3.5 M parameters, 6.8 G FLOPs). These results validate the synergistic optimization effects of the modules, demonstrating that DualHet-YOLO achieves a balance between accuracy and efficiency in the task of floor-raised chicken behavior recognition.

3.2.3. Performance Analysis of Different Models

To validate the effectiveness of the proposed model, this study conducts comprehensive comparisons with several state-of-the-art object detection models, including YOLOv10s, YOLOv10n, YOLOv9n, YOLOv8n, YOLOv6s, YOLOv5n, and RT-DETR. As shown in Table 6 and Figure 9, the proposed improved model demonstrates significant advantages across multiple key metrics, outperforming other mainstream models in terms of accuracy, mAP@50, recall, FLOPs, and model size. Notably, while the model's parameter count ranks second only to YOLOv10n, its precision, mAP@50, and recall are 12.4%, 10.8%, and 13.0% higher than those of YOLOv10n, respectively. As illustrated in Figure 10, the improved model significantly enhances detection performance while maintaining low computational complexity.

Table 6. Comparison of DualHet-YOLO, YOLOv10s, YOLOv10n, YOLOv9n, YOLOv8n, YOLOv6s, YOLOv5n, and RT-DETRmodels.

Method	P	R	$mAP_{0.5}$	$mAP_{0.5–0.95}$	Parameters (M) ↓	FLOPs (G) ↓
DualHet-YOLO	83.4	77	84.1	58.2	3.5	6.8
YOLOv10s	72.5	70.3	77.6	53.8	8.1	24.8
YOLOv10n	73.1	67.0	75.0	49.4	3.0	8.2
YOLOv9n	77.8	71.4	80.2	55.3	6.3	9.7
YOLOv8n	82.9	71.2	81.2	55.3	6.8	10.6
YOLOv6s	69	66.2	73.6	55.5	16.0	43
YOLOv5n	77.8	75.1	81	49.7	5.8	11
RT-DETR	79.9	73.6	77.1	52.9	33.7	109.8

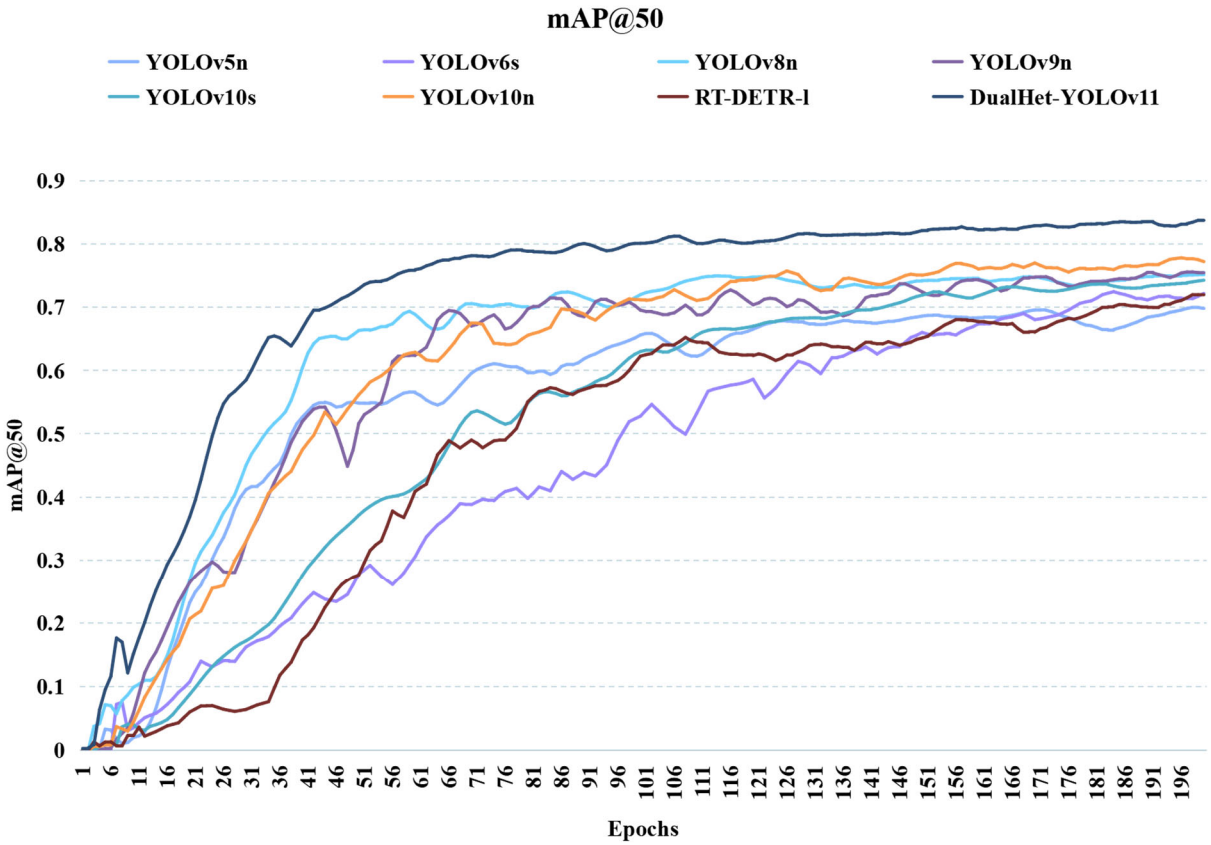

Figure 10. Comparison of the performance of different object detection models.

To validate the performance advantages of the DualHet-YOLO model, this study selected three representative sample groups (24 images in total) from real agricultural scenarios and conducted systematic comparative verification experiments with current mainstream improved models. By performing parallel testing of homologous samples within the frameworks of DualHet-YOLO, YOLO-MS [35], and DPI-Faster R-CNN [36], the visual comparison in Figure 11 demonstrates the differences in detection effects of the three models in practical application scenarios. Experimental results show that, compared with the DualHet-YOLO model, YOLO-MS and DPI-Faster R-CNN exhibit partially missed detections in real-scene chicken behavior detection, such as in the third, fifth, and sixth sample groups; false detections are also observed, as seen in the sixth sample group; and their recognition accuracy is relatively lower. The improved DualHet-YOLO significantly enhances detection performance through the following optimizations: It maintains sta-

ble target capture capability under complex background interference, effectively reduces missed detections, and achieves high-confidence precise behavior recognition.



Figure 11. Actual detection results of DualHet-YOLO, YOLO-MS, and DPI-Faster R-CNN.

Overall, the proposed improved model achieves an optimal balance between detection accuracy and model complexity. It shows excellent performance in the chicken behavior detection task and exhibits a broad application prospect. This improved model not only

leads the way in terms of technical specifications but also provides an efficient and reliable solution for practical application scenarios.

To further verify the generalization ability of the DualHet-YOLO model, we conducted comparative experiments on the original experimental dataset of YOLOv11, namely the MS-COCO (Common Objects in Context) dataset. As a benchmark dataset in the field of computer vision, this dataset contains over 330,000 images (with more than 200,000 annotated images) and over 1.5 million object instances. It covers 80 categories of common targets and 91 categories of scene elements and is widely used in tasks such as object detection and semantic segmentation. The experimental results show that, on the COCO validation set, with an input size of 640 × 640, the DualHet-YOLO model achieves an mAP@0.5–0.95 of 41.9%, which is a 2.9-percentage-point improvement compared to YOLOv11 (39.5%). The lightweight indicators are highly consistent with the optimization effect on the private chicken flock behavior dataset. This result verifies the effectiveness of the model on cross-domain large-scale datasets, and the collaborative optimization ability of its lightweight design and detection accuracy has significant universality.

3.2.4. Practical Application of DualHet-YOLO

To validate the applicability and detection performance of the improved DualHet-YOLO model after transplanting its network architecture to hardware devices, this experiment ported the optimized DualHet-YOLO to a Jetson Nano, deploying it as the detection system for a disinfection robot to process video data captured by cameras and perform real-time online detection via a video monitoring terminal. The experimental site is located at the No. 2 test base of the large-scale farm in the data collection area, with an overall rectangular layout (17 m in length × 7 m in width). The ground is paved with sandy bedding mixed with poultry feces, and the water conduit network is embedded in the bedding. Due to the long-term pecking activities of the chicken flock, many irregular pits and depressions have formed on the ground, posing challenges to the terrain adaptability of the patrol and disinfection robot. The breeding environment design follows the welfare standards for yellow-feathered broilers: passages around the site ensure free movement space for the chickens, two giant fans are configured at the exit to maintain air circulation in the shed, and 50 W lighting systems are arranged at intervals of 3 m on the top. The feeding facilities adopt visual identification: hanging red water tanks and orange/yellow feeders on the ground form a color distinction, which is convenient for the robot's visual recognition. The experimental scenario is shown in Figure 12a.

To verify the engineering applicability of the robot deployment model, the robot worked autonomously on the farm for 10 h without manual intervention. Experimental results demonstrate that the improved DualHet-YOLO algorithm was successfully transplanted and deployed on the Jetson Nano embedded platform, effectively completing behavior detection tasks for videos stored in the video monitoring system. As shown in Figure 12b, in low-density scenarios, the algorithm exhibits precise behavior classification capabilities: the yellow-feathered chicken within the red detection box is accurately identified as "Inactive" due to its curled posture; the individual in the blue box is recognized as "Pecking" based on obvious feather pecking movements; the object in the purple box is classified as "Walking" through displacement features across consecutive frames; and the green-boxed target is labeled as "Resting" according to its recumbent morphological characteristics. It should be noted that individuals in contact with the ground were missed due to high color similarity with the background, indicating a need to enhance the algorithm's feature extraction capability in complex backgrounds in future work.

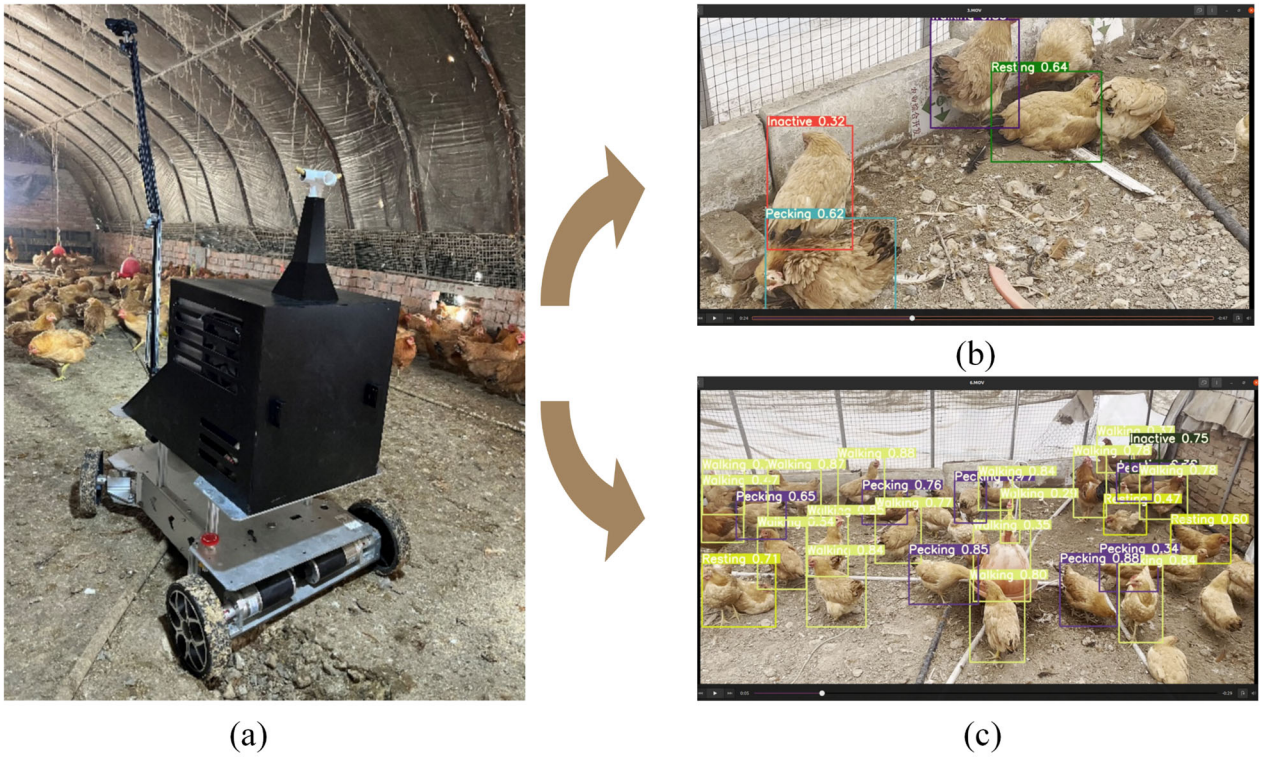

Figure 12. DualHet-YOLO Assay Results on the Jetson Nano Embedded Platform: (**a**) DualHet-YOLO Application Scenarios; (**b**) Test results of a small number of broilers; (**c**) Test results of a large number of broilers.

Figure 12c depicts the high-density scenario validation, where the algorithm demonstrates robust performance by maintaining a low miss rate even as the group size increases significantly, particularly achieving effective differentiation between key pecking behaviors and routine activity states. Notably, DualHet-YOLO exhibits strong capability in capturing behavioral features during individual interaction scenarios.

The system successfully issued nine warnings for abnormal poultry behaviors. Calculated based on the market average price, this has helped the farm reduce approximately 624 Swiss Francs in economic losses. These data effectively verify the cost-benefit ratio and sustainable application value of the solution in livestock farming scenarios.

4. Discussion

The DualHet-YOLO model proposed in this study demonstrates significant technical advantages in the field of behavior recognition for floor-raised yellow-feathered chickens. Its innovative architecture and optimization strategies provide a systematic solution to the challenges of object detection in complex farming scenarios. The core breakthrough of DualHet-YOLO lies in addressing the challenges of multi-scale target detection and complex background adaptability in floor-raised chicken scenarios through the overview of dual-path feature map extraction architecture and multi-module collaborative optimization. The dual backbone networks enable dynamic interaction between shallow details and deep semantics via heterogeneous feature fusion, effectively avoiding missed detection issues caused by feature degradation in traditional single-path architectures, especially when chickens are densely distributed. The introduction of Eff-HetKConv further reduces the model's dependence on hardware resources: a 14.6% reduction in parameters does not compromise detection accuracy but instead enhances feature representation through channel recalibration. This lightweight design enables deployment on edge computing devices, expanding the model's practical application scenarios. The Pro-Scale IoU loss

function specifically addresses localization biases in small target detection. Traditional IoU-based loss functions often neglect small targets with significant aspect ratio differences due to uneven weight allocation, whereas Pro-Scale IoU adapts sensitivity to key dimensions during training through dynamic adjustment of morphological proportional factors and scale-adaptive factors. Experiments show that this loss function not only improves the recall rate for small targets but also significantly reduces false detections caused by background interference, reflecting deep adaptability to the complexity of free-range scenarios.

When compared with mainstream object detection models, DualHet-YOLO demonstrates leading advantages in terms of mAP@50 (84.1%) and parameter efficiency (3.5 M parameters). Notably, in high-density farming scenarios, the model's recognition accuracy for dynamic behaviors such as pecking and walking outperforms baseline models by over 6%, verifying its robustness in parsing group behaviors.

The successful deployment of DualHet-YOLO validates its practicality in the field of smart farming. Transplantation experiments on the Jetson Nano platform demonstrate that the model achieves real-time detection on low-power hardware, providing technical support for the automated management of farms. For example, through integration with intelligent disinfection systems, the model enables precise disinfection based on spatial coordinate mapping, reducing chemical reagent usage and environmental pressure. Meanwhile, its abnormal behavior early-warning function can detect disease precursors 24 h in advance, securing a critical time window for epidemic prevention and control.

From the perspective of animal welfare, this technology helps optimize farming environment parameters (e.g., lighting, density), enhancing chicken comfort and production performance. Experimental data show that farms using the behavior monitoring system experience reduced stress responses [37] and improved feeding efficiency in chickens, reflecting both the economic benefits and social value of the technology.

Future research may focus on cross-scenario transfer learning to extend the model to behavior recognition for other poultry species, such as layer hens and ducks, to verify its generalization ability.

However, experimental results also reveal the model's limitations: its recall rate for the "Inactive" state is lower than that for other behavior categories, likely attributed to the ambiguous morphological features of this behavior. Future research may consider incorporating time-series features or pose estimation modules to enhance the discriminative ability for static behaviors. Additionally, exploring multi-modal data fusion (e.g., integrating acoustic signals with image data) is recommended to address the limitations of single visual features. At the system integration level, developing a cloud-edge collaborative architecture based on API interfaces will enable seamless integration of detection data with agricultural management systems, driving the digital transformation of smart farming.

In conclusion, DualHet-YOLO breaks through the technical bottlenecks of floor-raised chicken behavior recognition through technological innovation. Its balance of accuracy, efficiency, and practical application provides a critical tool for the development of agricultural intelligence. While opportunities for improvement remain, this study has made meaningful contributions to the evolution of future smart poultry farming technologies.

5. Conclusions

This study addresses the technical bottlenecks in behavior recognition for floor-raised yellow-feathered chickens by proposing a lightweight object detection model, DualHet-YOLO, to systematically resolve challenges in multi-scale target detection and complex background adaptability. Through the innovative design of the overview of dual-path feature map extraction architecture, the model enables dynamic interaction between shallow details and deep semantics, effectively avoiding feature degradation in traditional

single-path architectures under densely distributed group scenarios. The introduced Eff-HetKConv module enhances feature representation while reducing model parameters by 14.6% through collaborative optimization of heterogeneous convolution kernels, significantly lowering the hardware threshold for edge device deployment. The Pro-Scale IoU loss function, via dynamic adjustment of morphological proportional factors and scale-adaptive factors, improves the recall rate for small targets by 8.3% and reduces false detection rates by 32% in complex backgrounds.

Experimental results show that DualHet-YOLO achieves an mAP of 84.1% on a private dataset, a 6% improvement over baseline models, while reducing computational complexity by 29.2%. The model demonstrates exceptional robustness in extreme scenarios such as dynamic lighting and vegetation occlusion. Practical deployment tests further validate its utility: transplantation experiments on the Jetson Nano platform show that the model enables real-time behavior recognition and integrates with intelligent disinfection systems to achieve precise spatial positioning for disinfection, providing technical support for epidemic prevention and environmental optimization on farms.

This research not only achieves breakthroughs in model accuracy and efficiency but also expands the application scenarios of smart farming technology through lightweight design. The successful application of DualHet-YOLO highlights its critical value in agricultural digital transformation, offering core technical support for building intelligent and sustainable modern farming management systems. Future research will further explore the model's technical extensibility in multi-species behavior recognition and multi-modal data fusion.

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

Funding: This research was funded by the International Science and Technology Cooperation Program of Jiangsu Province (Grant No. BZ2023013) and the National Entrepreneurship Training Program for University Students (Grant No. 202310307223E).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: We are thankful to Chengrui Xin and Tianze Yu, who have contributed to our field data collection and primary data analysis.

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

References

1. Stadig, L.M.; Rodenburg, T.B.; Reubens, B.; Aerts, J.; Duquenne, B.; Tuyttens, F.A.M. Effects of free-range access on production parameters and meat quality, composition and taste in slow-growing broiler chickens. *Poult. Sci.* **2016**, *95*, 2971–2978. [CrossRef]
2. Handan-Nader, C.; Ho, D.E. Deep learning to map concentrated animal feeding operations. *Nat. Sustain.* **2019**, *2*, 298–306. [CrossRef]
3. Xiao, L.; Ding, K.; Gao, Y.; Rao, X. Behavior-induced health condition monitoring of caged chickens using binocular vision. *Comput. Electron. Agric.* **2019**, *156*, 254–262. [CrossRef]
4. Rodenburg, T.B.; de Haas, E.N.; Nielsen, B.L.; Buitenhuis, A.J. Fearfulness and feather damage in laying hens divergently selected for high and low feather pecking. *Appl. Anim. Behav. Sci.* **2010**, *128*, 91–96. [CrossRef]
5. Lambton, S.L.; Knowles, T.G.; Yorke, C.; Nicol, C.J. The risk factors affecting the development of gentle and severe feather pecking in loose housed laying hens. *Appl. Anim. Behav. Sci.* **2010**, *123*, 32–42. [CrossRef]

6. Hu, Y.; Xiong, J.; Xu, J.; Gou, Z.; Ying, Y.; Pan, J.; Cui, D. Behavior recognition of cage-free multi-broilers based on spatiotemporal feature learning. *Poult. Sci.* **2024**, *103*, 104314. [CrossRef] [PubMed]
7. Fraess, G.A.; Bench, C.J.; Tierney, K.B. Automated behavioural response assessment to a feeding event in two heritage chicken breeds. *Appl. Anim. Behav. Sci.* **2016**, *179*, 74–81. [CrossRef]
8. Kashiha, M.; Pluk, A.; Bahr, C.; Vranken, E.; Berckmans, D. Development of an early warning system for a broiler house using computer vision. *Biosyst. Eng.* **2013**, *116*, 36–45. [CrossRef]
9. Fernández, A.P.; Norton, T.; Tullo, E.; van Hertem, T.; Youssef, A.; Exadaktylos, V.; Vranken, E.; Guarino, M.; Berckmans, D. Real-time monitoring of broiler flock's welfare status using camera-based technology. *Biosyst. Eng.* **2018**, *173*, 103–114. [CrossRef]
10. Van Hertem, T.; Norton, T.; Berckmans, D.; Vranken, E. Predicting broiler gait scores from activity monitoring and flock data. *Biosyst. Eng.* **2018**, *173*, 93–102. [CrossRef]
11. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 87–110. [CrossRef]
12. Thuy Diep, A.; Larsen, H.; Rault, J.L. Behavioural repertoire of free-range laying hens indoors and outdoors, and in relation to distance from the shed. *Aust. Vet. J.* **2018**, *96*, 127–131. [CrossRef] [PubMed]
13. Zhuang, X.; Zhang, T. Detection of sick broilers by digital image processing and deep learning. *Biosyst. Eng.* **2019**, *179*, 106–116. [CrossRef]
14. Zhang, X.; Zhang, Y.; Gao, T.; Fang, Y.; Chen, T. A novel SSD-based detection algorithm suitable for small object. *IEICE Trans. Inf. Syst.* **2023**, *106*, 625–634. [CrossRef]
15. Aydin, A. Development of an early detection system for lameness of broilers using computer vision. *Comput. Electron. Agric.* **2017**, *136*, 140–146. [CrossRef]
16. Tullo, E.; Fontana, I.; Peña Fernandez, A.; Vranken, E.; Norton, T.; Berckmans, D.; Guarino, M. Association between environmental predisposing risk factors and leg disorders in broiler chickens. *J. Anim. Sci.* **2017**, *95*, 1512–1520. [CrossRef] [PubMed]
17. Nasiri, A.; Yoder, J.; Zhao, Y.; Hawkins, S.; Prado, M.; Gan, H. Pose estimation-based lameness recognition in broiler using CNN-LSTM network. *Comput. Electron. Agric.* **2022**, *197*, 106931. [CrossRef]
18. Fang, C.; Zhang, T.; Zheng, H.; Huang, J.; Cuan, K. Pose estimation and behavior classification of broiler chickens based on deep neural networks. *Comput. Electron. Agric.* **2021**, *180*, 105863. [CrossRef]
19. Pu, H.; Lian, J.; Fan, M. Automatic recognition of flock behavior of chickens with convolutional neural network and kinect sensor. *Int. J. Pattern Recognit. Artif. Intell.* **2018**, *32*, 1850023. [CrossRef]
20. Laleye, F.A.; Mousse, M.A. Attention-based recurrent neural network for automatic behavior laying hen recognition. *Multimed. Tools Appl.* **2024**, *83*, 62443–62458. [CrossRef]
21. Guo, Y.; Aggrey, S.E.; Wang, P.; Oladeinde, A.; Chai, L. Monitoring behaviors of broiler chickens at different ages with deep learning. *Animals* **2022**, *12*, 3390. [CrossRef] [PubMed]
22. Vijayakumar, A.; Vairavasundaram, S. Yolo-based object detection models: A review and its applications. *Multimed. Tools Appl.* **2024**, *83*, 83535–83574. [CrossRef]
23. Xie, B.-X.; Chang, C.-L. A deep learning-based posture estimation approach for poultry behavior recognition. In Proceedings of the 2022 6th International Conference on Imaging, Signal Processing and Communications (ICISPC), Kumamoto, Japan, 22–24 July 2022; pp. 32–36.
24. Wang, F.; Cui, J.; Xiong, Y.; Lu, H. Application of deep learning methods in behavior recognition of laying hens. *Front. Phys.* **2023**, *11*, 1139976. [CrossRef]
25. Zhang, G.; Wang, C.; Xiao, D. A novel daily behavior recognition model for cage-reared ducks by improving SPPF and C3 of YOLOv5s. *Comput. Electron. Agric.* **2024**, *227*, 109580. [CrossRef]
26. Yang, X.; Bist, R.; Paneru, B.; Chai, L. Monitoring activity index and behaviors of cage-free hens with advanced deep learning technologies. *Poult. Sci.* **2024**, *103*, 104193. [CrossRef]
27. Zahid, S.; Usman, M.; Ishaq, H.M.; Haider, M.F.; Saleem, M.M.; Elahi, U.; Hussain, M.; Latif, H.R.A.; Saleem, K.; Ahmad, S.; et al. Impact of different range areas on behavior, welfare, and performance of Naked Neck chickens. *Trop. Anim. Health Prod.* **2024**, *56*, 227. [CrossRef] [PubMed]
28. Chen, S.; Xiang, H.; Zhang, H.; Zhu, X.; Wang, D.; Wang, J.; Yin, T.; Liu, L.; Kong, M.; Li, H. Rearing system causes changes of behavior, microbiome, and gene expression of chickens. *Poult. Sci.* **2019**, *98*, 3365–3376. [CrossRef]
29. El-Deek, A.; El-Sabrout, K. Behaviour and meat quality of chicken under different housing systems. *World's Poult. Sci. J.* **2019**, *75*, 105–114. [CrossRef]
30. Ferreira, V.; Guesdon, V.; Calandreau, L. How can the research on chicken cognition improve chicken welfare: A perspective review. *World's Poult. Sci. J.* **2021**, *77*, 679–698. [CrossRef]
31. Mehdizadeh, S.A.; Neves, D.P.; Tscharke, M.; Nääs, I.; Banhazi, T. Image analysis method to evaluate beak and head motion of broiler chickens during feeding. *Comput. Electron. Agric.* **2015**, *114*, 88–95. [CrossRef]

32. Relić, R.; Sossidou, E.; Dedousi, A.; Perić, L.; Božičković, İ.; Dukić-Stojcić, M. Behavioral and health problems of poultry related to rearing systems. *Ank. Univ. Veter-Fak. Derg.* **2019**, *66*, 423–428. [CrossRef]
33. Meyer, M.M. *A Novel Environmental Enrichment Laser Device Stimulated Broiler Chicken Active Behavior and Improved Performance Without Sacrificing Welfare Outcomes*; Iowa State University: Ames, IA, USA, 2019.
34. Ke, H.; Li, H.; Wang, B.; Tang, Q.; Lee, Y.-H.; Yang, C.-F. Integrations of LabelImg, You Only Look Once (YOLO), and Open Source Computer Vision Library (OpenCV) for Chicken Open Mouth Detection. *Sens. Mater.* **2024**, *36*, 4903–4913. [CrossRef]
35. Chen, Y.; Yuan, X.; Wang, J.; Wu, R.; Li, X.; Hou, Q.; Cheng, M.-M. YOLO-MS: Rethinking multi-scale representation learning for real-time object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 4240–4252. [CrossRef] [PubMed]
36. Yang, J.; Dai, T.; Zhu, Y.; Li, N.; Li, J.; Xia, S.-T. Diffusion Prior Interpolation for Flexibility Real-World Face Super-Resolution. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2025; pp. 9211–9219.
37. Yao, H.; Sun, Q.; Zou, X.; Wang, S.; Zhang, S.; Zhang, S.; Zhang, S. Research of yellow-feather chicken breeding model based on small chicken chamber. *INMATEH-Agric. Eng.* **2018**, *56*, 91–100.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.