PCN-YOLO：AN IMPROVED LION-HEAD GOOSE DETECTION METHOD BASED ON YOLO11

JIANHAO JIANG[1], ZHIGUO DU[1], BIN WEN[1], ZHIHUI WU[1], XUDONG LIN[1]

[1]College of Mathematics and Information, South China Agricultural University, Guangzhou 510642, China
E-MAIL: jiangjianhao@stu.scau.edu.cn, Bluetooth@scau.edu.cn,
wenzip@scau.edu.cn,wuzhihui@scau.edu.cn,hunanlxd@163.com

Abstract:

With the continuous advancement of computer vision and deep learning, object detection has achieved remarkable progress across diverse domains, yet specific scenarios such as lion-head goose detection in complex backgrounds still face significant challenges due to issues like object overlap and occlusion, scale variations, and environmental interference. To tackle these challenges, this paper presents PCN-YOLO, an improved detection method based on YOLO11. The proposed approach first replaces the original C3k2 module in the backbone with a Poly Kernel Inception (PKI) block, enabling effective extraction of multi-scale information from input feature maps. Subsequently, a newly designed Context-Guided Attention Fusion (CGAFusion) module at the network neck fuses contextual information to precisely identify lion-head goose positions and features. Finally, replacing the conventional loss function with the Normalized Wasserstein Distance (NWD) loss function strengthens object contour capture. Experimental results demonstrate that PCN-YOLO outperforms the YOLO11n model with improvements of 1.5% in precision, 2.3% in recall, 1.7% in mAP@50, and 1.9% in mAP@50-95, achieving outstanding performance metrics of 0.896, 0.852, 0.918, and 0.567 respectively, thus validating its practical effectiveness.

Keywords:

Computer Vision; Object Detection; Multi-scale Feature Extraction; YOLO11; Lion-head Goose

1. Introduction

In recent years, the lion-head goose industry has emerged as a pivotal sector in China's agriculture, particularly in Chenghai District, Shantou, where a complete industrial chain encompassing goose breeding, hatchling sales, meat goose rearing, and marinated processing has been established[1]. Preliminary statistics indicate that by 2024, the annual sales volume of lion-head geese in Chenghai District is expected to reach 10 million, with the full industrial chain generating over 6 billion yuan in revenue[2]. For lion-head goose farming, non-invasive object detection technology is beneficial for enabling precision management without disturbing them[3]. By accurately detecting goose quantity and growth status, farmers can optimize feed ration, improve breeding environments, and enhance economic efficiency. Additionally, analyzing goose behavior through object detection provides insights into their living habits, supporting scientific farming practices. Thus, applying advanced object detection technology to lion-head goose farming holds significant practical value, driving the industry toward intelligent and modern development.

Despite the rapid advancements in deep learning and object detection, their applications in lion-head goose detection remain limited, with most existing studies focusing on poultry such as chickens and ducks. In 2024, Xiao et al. proposed DHSW-YOLO[4], a real-time detection model for white-feathered Muscovy ducks under varying lighting conditions. By simplifying the detection head of YOLOv8 and integrating SENet attention and WIoU v3 loss, DHSW-YOLO achieved a 2.2% increase in mAP (from 92.2% to 94.4%), reduced model size by 2.8 MB, and accelerated inference time by 1.2 ms compared to the original YOLOv8. Ji et al. (2024) developed YOLO-FSG for chicken feeding behavior recognition[5], introducing C2F-FEblock to enhance feature extraction and reduce complexity. By replacing traditional convolutions and C2F modules with GSConv (group shuffle convolution) and VovGSCSP, and optimizing the detection head with parameter-sharing grouped convolutions, YOLO-FSG achieved an mAP@0.5 of 97.1% with 1.94 M parameters and 4 GFLOPs, outperforming YOLOv5n, YOLOv7n, and YOLOv8n. Pei et al. (2025) improved YOLOv8n for real-time dead chicken detection in cage farming[6], incorporating cross stage partial hetconv and CSPHet in the backbone, SEAM in the Neck, and DySample upsampling. Their method achieved an mAP of 95.8% with 2.46 MB parameters, representing a 1.5% accuracy improvement and 18.3% parameter reduction compared to YOLOv8n.

To address challenges specific to lion-head goose detection, including free-range environments, complex backgrounds, significant scale variations, and mutual

occlusion, this paper proposes PCN-YOLO, an optimized YOLO11-based method. The contributions are threefold: (1) Replacing the original C3k2 module with a PKI block parallelizes convolution kernels of varying sizes, significantly improving multi-scale feature extraction for lion-head geese across different poses and scales. (2) Introducing a CGAFusion module at the network neck enhances contextual information integration, improving target-background discriminability and reducing misjudgment. (3) Adopting the NWD loss function strengthens contour and edge representation, improving detection stability. Experimental results demonstrate that PCN-YOLO outperforms YOLO11n in precision (1.5% increase), recall (2.3% increase), mAP@50 (1.7% increase), and mAP@50-95 (1.9% increase), validating its effectiveness in complex lion-head goose detection scenarios.

2. Dataset

The dataset used in this study is derived from the publicly available Lion-head Goose Dataset by Yuhong Feng et al.[7], which was collected via cameras deployed at a goose farm in Chenghai District, Shantou, Guangdong Province, China. The dataset contains two subsets: "large geese" and "small geese". This paper focuses on the large goose subset, consisting of 1,949 images. These images were divided into training, validation, and test sets at a ratio of 7:2:1, resulting in 1,368, 392, and 189 images, respectively.

To enhance the model's generalization and robustness, a series of data augmentation techniques were applied to the training set, including horizontal flipping, rotation, brightness adjustment, blurring, and noise addition. These augmentations expanded the training set to 2,736 images, with a total dataset size of 3,317 images. Figure 1 illustrates representative samples from the dataset.

FIGURE 1. Example of the dataset

3. Research Methodology

3.1 YOLO11

YOLO (You Only Look Once) [8], a groundbreaking innovation in object detection, transforms the task into a regression problem, enabling end-to-end direct prediction and significantly simplifying the detection pipeline. Since its inception, the YOLO series has undergone multiple iterations, continuously advancing object detection technology[9].The baseline model YOLO11 used in this study, as as shown in Figure 2, retains the classic backbone-neck-head architecture of YOLO while integrating C3k2 and C2PSA modules into the backbone and neck structures [10]. This integration maintains high performance while reducing computational requirements, making it suitable for resource-constrained devices.

The C3k2 module (Figure 2(e)) processes input feature maps through an initial convolution layer for low-level feature extraction, followed by splitting into regions and multiple repeated C3 blocks. The C3 block efficiently handles redundant gradient residuals and enhances information flow between dense blocks[11], balancing inference speed and detection accuracy. Feature maps are concatenated via a Contact layer and further processed by a Convolution-BatchNorm-SiLU(CBS) layer for final integration.

The C2PSA module (Figure 2(j)) begins with a CBS convolution layer, splitting the feature maps into two branches. one preserves raw information, while the other undergoes PSABlock processing for feature reinforcement (Figure 2(h)). The two branches are concatenated and passed through another CBS layer to output fused multi-scale features. This design optimizes multi-dimensional feature extraction and fusion, improving network expressiveness.

Leveraging YOLO11's advantages, this study proposes PCN-YOLO, an enhanced model integrating the PKI block, CGAFusion module, and NWD loss function. The architecture of PCN-YOLO is detailed in Figure 3.

3.2 PKI Block

The PKI (Poly Kernel Inception) block was introduced by Xinhao Cai et al.[12]. This block consists of the PKI module and the Context Anchor Attention (CAA) module, as depicted in Figure 4(c). The design of the PKI module is inspired by an in - depth analysis of features of targets at different scales. Its core idea is to process the input feature map with convolution kernels of different sizes to extract multi-

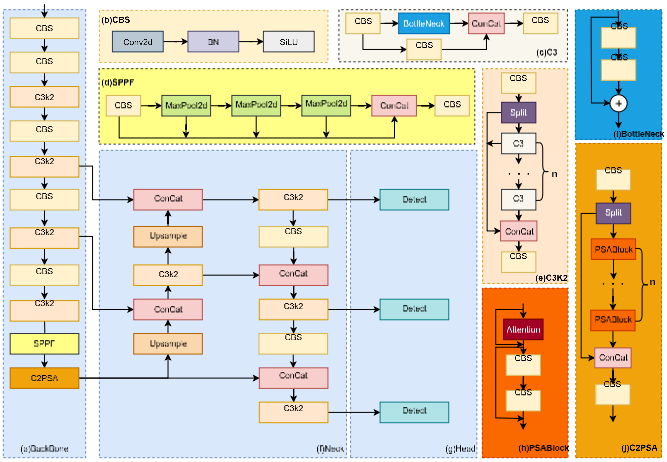

FIGURE 2. YOLO11 Network architecture

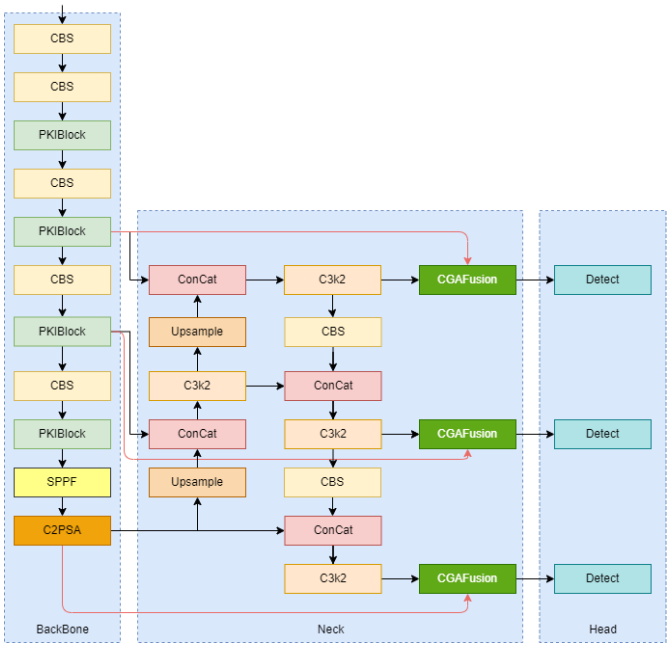

FIGURE 3. PCN-YOLO Network architecture

scale feature information.

As shown in Figure 4(a), the PKI Module first performs multi scale convolution operations on the input feature map. Small convolution kernels, such as 3×3 ones, have a strong ability to extract local features and can capture fine textures and detailed information of the target. For example, for the fine features of lion - head geese like feather textures and eyes, small convolution kernels can clearly extract them, providing accurate local information for subsequent target recognition.

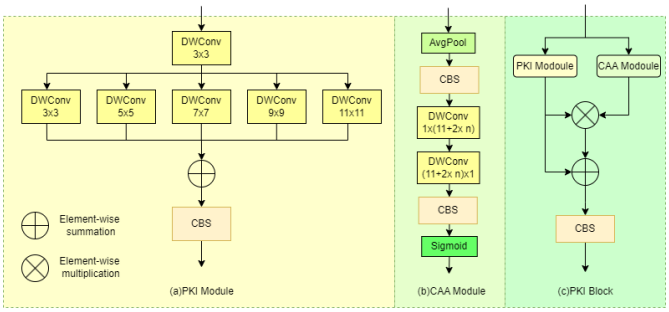

FIGURE 4. Structure of PKI

And the parallel depth - separable convolution kernels of different sizes, such as 5×5 and 7×7, can capture context information within a larger receptive field. When detecting a group of relatively close and large lion - head geese, these large convolution kernels can focus on large features like the heads and bodies of the geese, compensating for the lack of the small convolution kernels in extracting global information.

The PKI block also integrates the CAA Module, as shown in Figure 4(b). The CAA module is an attention - mechanism component that fuses context awareness. It realizes refined feature weighting by integrating global and local features. Specifically, the module first captures the global statistical information of the features through average pooling(AvgPool) to extract the overall semantic context. Then, it uses two depth - separable convolutions(DWConv) to extract local detailed features in the horizontal and vertical directions respectively, enriching the dimension of feature representation. Finally, it generates attention weights through a 1×1 convolution and a Sigmoid activation function.

3.3 CGAFusion Module

The CGAFusion module, proposed by Zixuan Chen *et al.*[13], adaptively fuses low-level and high-level features by learning spatial and channel-wise weights to modulate feature responses. In PCN-YOLO, the CGAFusion Module serves as the core processing unit, first performing additive fusion on input features F_{low} (low-level features) and F_{high} (high-level features), followed by parallel SpatialAtt (Spatial Attention) and ChannelAtt (Channel Attention) modules to mine feature saliency in spatial and channel dimensions. The SpatialAtt Module extracts spatial statistical information via global average pooling (GAP) across spatial dimensions and global max pooling (GMP) across channel dimensions, generating spatial attention weights through ConCat (Concatenation) and CBS layer processing to precisely locate critical spatial regions of lion-head geese. The ChannelAtt Module aggregates global channel information using GAP across channel dimensions,

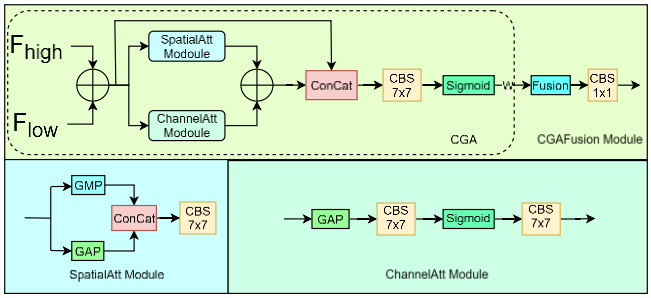

FIGURE 5. Structure of CGAFusion

then generates channel attention weights via a CBS layer and Sigmoid activation to select discriminative channels for goose detection. After dual-attention processing, the fused features undergo secondary fusion, CBS layer processing, Sigmoid activation, and final Fusion operations, with a 1×1 CBS convolution adapting feature dimensions to produce high-quality representations balancing spatial localization accuracy and channel semantic discriminability for lion-head goose detection.The feature fusion formula is expressed as:

$$F = F_{\text{low}} + F_{\text{high}} + F_{\text{low}} \times w + F_{\text{high}} \times (1 - w) \quad (1)$$

where F_{low} and F_{high} denote low-level and high-level inputfeatures, respectively, and w represents attention weights output by the CGA module.

Traditional fusion methods like simple concatenation or addition neglect semantic differences and contextual information between feature levels, combining features without adaptive weighting and leading to background interference in complex scenes. In contrast, the CGAFusion module leverages contextual information and dual spatial-channel attention to intelligently fuse features, improving adaptability to complex environments. For lion-head goose detection in cluttered farming scenes, CGAFusion accurately identifies goose positions and features using contextual cues, minimizing background misjudgment and enhancing accuracy.

3.4 NWD loss

In object detection tasks, the design of bounding box regression loss directly affects the performance of the detection model. Traditional IoU-based losses (such as GIoU, DIoU, CIoU) mainly focus on the overlap degree between the predicted box and the ground-truth box, while ignoring the differences in spatial position and scale between the two, which may lead to insufficient adaptability to objects with scale changes. In addition, when there is no intersection between the predicted box and the ground-truth box, the gradient of IoU will disappear, affecting the optimization process. Therefore, we introduce the Normalized Wasserstein Distance (NWD)[14] into the regression loss to measure the matching degree of the spatial position and scale between the predicted box and the ground-truth box, thereby improving the detection accuracy.

In this study, let the predicted bounding box $Bp = (x_{\text{p1}}, y_{\text{p1}}, x_{\text{p2}}, y_{\text{p2}})$ and the ground-truth bounding box $Bt = (x_{\text{t1}}, y_{\text{t1}}, x_{\text{t2}}, y_{\text{t2}})$, where (x_1, y_1) and (x_2, y_2) represent the coordinates of the top-left and bottom-right corners of the bounding box, respectively. First, calculate the IoU loss:

$$L_{\text{IoU}} = (1 - IoU) \cdot w \quad (2)$$

where IoU is the intersection-over-union of the predicted box and the ground-truth box, and w is the object score weighting coefficient, making the contribution of high-confidence objects to the loss greater.

To enhance the constraint of the loss function on the geometric matching of the bounding box, we further calculate the Wasserstein-2 distance between the predicted box and the ground-truth box. This distance consists of two parts,the Euclidean distance of the center point and the scale difference. The Euclidean distance of the center point is calculated as follows:

$$d_{\text{center}} = (x_{\text{p,center}} - x_{\text{t,center}})^2 + (y_{\text{p,center}} - y_{\text{t,center}})^2 + \epsilon \quad (3)$$

where ϵ is a numerical stability factor, and $(x_{\text{p,center}}, y_{\text{p,center}})$ and $(x_{\text{t,center}}, y_{\text{t,center}})$ are the center coordinates of the predicted box and the ground-truth box, respectively:

$$x_{p,\text{center}} = \frac{x_{p1}+x_{p2}}{2}, \quad y_{p,\text{center}} = \frac{y_{p1}+y_{p2}}{2} \quad (4)$$

$$x_{t,\text{center}} = \frac{x_{t1}+x_{t2}}{2}, \quad y_{t,\text{center}} = \frac{y_{t1}+y_{t2}}{2} \quad (5)$$

The scale difference is measured by calculating the width and height differences between the predicted box and the ground-truth box:

$$d_{\text{wh}} = \frac{(w_p - w_t)^2 + (h_p - h_t)^2}{4} \quad (6)$$

where w and h are the width and height of the bounding box, respectively:

$$w_p = x_{p2} - x_{p1} + \epsilon, \quad h_p = y_{p2} - y_{p1} + \epsilon \quad (7)$$

$$w_t = x_{t2} - x_{t1} + \epsilon, \quad h_t = y_{t2} - y_{t1} + \epsilon \quad (8)$$

Finally, the Wasserstein-2 distance is defined as:

$$d_{\text{W}}^2 = d_{\text{center}} + d_{\text{wh}} \quad (9)$$

According to the definition of the Wasserstein distance, the Wasserstein loss function is constructed:

$$L_{\text{W}} = \exp\left(-\frac{\sqrt{d_{\text{W}}^2}}{C+\epsilon}\right) \quad (10)$$

where C is a scaling factor (in this paper, C=12.8), used to adjust the scale range of the loss, enabling it to optimize stably during the training process. Experiments show that the Wasserstein loss can effectively improve the model's adaptability to objects with large scale changes.

In addition, to further optimize the prediction accuracy of

the bounding box, the Distribution Focal Loss (DFL) is introduced into the loss function. The core idea of DFL is: in object box prediction, the left, right, top, and bottom boundaries of the bounding box can be expressed as a discrete distribution relative to anchor points. Therefore, the ground-truth box is converted into a discrete distribution, and the error between the predicted distribution and the ground-truth distribution is calculated:

$$L_{\mathrm{DFL}} = \mathrm{DFL}(P_{\mathrm{dist}}, T_{\mathrm{ltrb}}) \cdot w \tag{11}$$

where P_{dist} is the predicted boundary distribution, and T_{ltrb} is the discrete target distribution after the ground-truth box is converted.

The final object box regression loss consists of the IoU loss, Wasserstein loss, and DFL loss:

$$L_{NWD} = \lambda_{\mathrm{IoU}} L_{\mathrm{IoU}} + (1 - \lambda_{\mathrm{IoU}}) L_{\mathrm{W}} + L_{\mathrm{DFL}} \tag{12}$$

where λ_{IoU} controls the relative contribution of the IoU loss and the Wasserstein loss. In this paper, $\lambda_{\mathrm{IoU}} = 0.2$ is set to ensure that the Wasserstein loss plays a more important role in the regression task.

Experimental results show that compared with traditional IoU and its variant losses (such as GIoU, DIoU, CIoU)[15], the proposed Wasserstein regression loss can achieve higher bounding box localization accuracy in detection tasks, especially performing exceptionally well in scenarios with large object scale changes. For example, in the lion-head goose detection task, this method can more accurately capture the object contour, improving the stability and robustness of detection.

4. Experimental Results and Discussions

4.1 Experimental Platform and Evaluation Metrics

All experiments in this paper were conducted using the cloud servers of Parallel Intelligent Computing Cloud Company. The development framework was PyTorch 2.2.0 and Ubuntu 22.04. The server was configured with an RTX 4090 GPU with 24GB of video memory and a 10v AMD EPYC 7402 CPU. The training parameters were set as follows: the number of epochs was set to 300, the batch - size was set to 32, the image input size was 640, and the optimizer used for training was Adaptive Moment Estimation (Adam).

In the lion - head goose target detection task, evaluation metrics are important for measuring the performance of the model. The number of parameters refers to the total number of trainable parameters in the model, which determines the model's storage requirements and complexity. The computational complexity (GFLOPs, Giga Floating Point Operations) measures the number of floating - point operations performed by the model during the inference process, which determines the model's computational cost. Precision (P) and Recall (R) are core metrics in target detection, used to evaluate the model's detection accuracy and recall ability. Precision measures the proportion of detected lion - head goose targets that are actually lion - head geese, while Recall measures the proportion of all lion - head goose targets that are successfully detected. Their calculation formulas are as follows:

$$P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN} \tag{13}$$

TP (True Positives) represents the number of correctly detected lion - head goose targets, FP (False Positives) represents the number of falsely detected negative samples, that is, the number of times the model incorrectly identifies the background or other objects as lion - head geese, and FN (False Negatives) represents the number of lion - head goose targets that fail to be detected, that is, the number of real lion - head goose targets that the model fails to recognize. High Precision means a low misclassification rate of detected targets, while high Recall means that more real targets are successfully recognized. However, there is usually a trade - off between the two. An increase in Precision may lead to a decrease in Recall, and vice versa.

To more comprehensively evaluate the performance of the detection model, Average Precision (AP) and mean Average Precision (mAP) are commonly used. AP measures the area under the Precision - Recall curve, representing the model's average precision at different recall rates. Its calculation formula is as follows:

$$AP = \int_0^1 P(R)\, dR, \quad mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{14}$$

where N is the total number of target categories. In the lion - head goose target detection task, N is 1, meaning that mAP is calculated only for the lion - head goose category. mAP@50 (mAP@50) represents the mAP when the IoU threshold is fixed at 0.5, while mAP@[.50:.95] (mAP@50 - 95) is the mean value calculated at different IoU thresholds from 0.5 to 0.95 (with a step size of 0.05), which can more comprehensively measure the detection accuracy of the detector under different matching criteria.

4.2 Ablation experiments

To validate the contributions of the proposed PKI, CGAFusion, and NWD loss function to lion-head goose detection, ablation experiments were conducted. By sequentially removing each module and comparing results under identical experimental conditions and datasets, the independent effects and combined performance of each component were analyzed. The experimental results are shown in the table 1.

The baseline model achieved P of 88.1%, R of 82.9%, mAP@50 of 90.1%, and mAP@50-95 of 54.8%. Introducing PKI improved P to 89.1%, R to 83.8%, mAP@50 to 91.2%,

TABLE 1. Ablation experiments

PKI	CGAFusion	NWD_loss	Parameters	GFLOPs	P	R	mAP@50	mAP@50-95
×	×	×	2,582,347	6.3	0.881	0.829	0.901	0.548
√	×	×	2,582,787	7.7	0.891	0.838	0.912	0.556
×	√	×	2,970,532	8.7	0.888	0.836	0.907	0.557
×	×	√	2,582,347	6.3	0.891	0.83	0.906	0.546
√	√	×	2,970,972	10.0	0.892	0.848	0.914	0.564
√	√	√	2,970,972	10.0	**0.896**	**0.852**	**0.918**	**0.567**

and mAP@50-95 to 55.6%. This indicates PKI enhances target discriminability through improved feature representation, reducing false positives and missed detections. Adding only CGAFusion increased P to 88.8%, R to 83.6%, mAP@50 to 90.7%, and mAP@50-95 to 55.7%, demonstrating its ability to stabilize multi-scale detection via cross-scale global feature aggregation. Using only NWD loss improved P to 89.1%, R to 83%, and mAP@50 to 90.6%, optimizing bounding box matching quality and reducing degradation issues through Wasserstein distance minimization.

The combination of PKI and CGAFusion further boosted P, R, mAP@50 and mAP@50-95 to 89.2%, 84.8%, 91.4%, and 56.4% respectively. Finally, integrating all three components achieved optimal performance, increasing P, R, mAP@50, and which outperforms the YOLOv8n model by 1.5%, 2.3%, 1.7%, and 1.9% in these metrics.

These results confirm that PKI strengthens feature extraction, CGAFusion enhances multi-scale adaptability, and NWD improves localization accuracy. Their combined implementation significantly advances lion-head goose detection in complex farming environments.

4.3 comparison with other networks

To validate the effectiveness of the method proposed in this paper, comparative experiments were conducted among YOLOv8n, YOLOv9t, YOLOv10n, YOLO11n, YOLO12n, and the PCN - YOLO proposed in this paper. The evaluation metrics included the number of parameters, computational complexity, Precision (P), Recall (R), mAP@50, and mAP@50-95. The experimental results show that YOLO11n has the best overall performance among the existing models, so it was selected as the baseline model. Compared with YOLO11n, PCN - YOLO increases P by 1.5%, R by 2.3%, mAP@50 by 1.7%, and mAP@50-95 by 1.9% with little change in the number of parameters and computational complexity. This validates the effectiveness of the PKI, CGAFusion, and NWD loss functions, enabling the model to achieve better performance in the lion - head goose target detection task. The comparative experimental results are shown in Table 2.

TABLE 2. Comparative experiments

	Parameters	GFLOPs	P	R	mAP@50	mAP@50-95
YOLOv8n	2,684,563	6.8	0.879	0.823	0.899	0.535
YOLOv9t	1,730,019	6.4	0.878	0.831	0.899	0.536
YOLOv10n	2,265,363	6.5	0.869	0.83	0.902	0.538
YOLO11n	2,582,347	6.3	0.881	0.829	0.901	0.548
YOLO12n	2,556,923	6.3	0.874	0.832	0.901	0.545
PCN-YOLO	2,970,972	10.0	**0.896**	**0.852**	**0.918**	**0.567**

4.4 visualization of experimental results

To visually demonstrate the performance of different models in lion-head goose detection, this paper selects four representative samples for visualization analysis. The results are presented in Figure 6, where the first row displays lion-head goose detection results from the YOLO11n model and the second row shows PCN-YOLO outputs. Here, blue bounding boxes represent model detections, yellow bounding boxes highlight missed targets, and red bounding boxes denote false positive cases. Although YOLO11n accurately detects targets under most conditions, it still exhibits missed detections and localization errors in scenarios with dense geese, occlusion, or blurred boundaries. By contrast, PCN-YOLO more accurately identifies overlapping geese and reduces missed detections and false positives. These visual results further validate the effectiveness of PCN-YOLO in addressing real-world challenges in goose farming environments.

FIGURE 6. Comparison of YOLO11n and PCN-YOLO results

5. Conclusions and Future Work

In this paper, we propose the PCN-YOLO model for lion-head goose detection, which significantly improves detection accuracy and localization capabilities by integrating the PKI block,CGAFusion and NWD loss function into the YOLO11n framework.Ablation and comparative experiments demonstrate that PCN-YOLO outperforms existing object detection models across multiple metrics, particularly showcasing robust performance and higher accuracy in complex scenarios involving dense targets and occlusion. Additionally, visualization results validate the model's effectiveness in real-world applications, with notable improvements in precision, recall, and localization accuracy.

For future research, we aim to optimize computational efficiency and explore lightweight network architectures to enable real-time detection on resource-constrained devices. Given the practical demands of lion-head goose detection, we will also enhance the model's generalization to diverse environmental conditions, including varying lighting, weather, and dynamic scenes. Furthermore, comprehensive evaluations using expanded datasets will be conducted to validate the broad applicability of our approach across different domains.

References

[1] Yu, S. J., & Leng, M. Q. (2024). Study on the high-quality development path of Chenghai Shitou goose industry in Shantou. Modern Business, (22), 159-163.

[2] Cai, X. D. (2024, December 28). Building a brand to create a 10-billion-yuan lion-head goose industry cluster. Shantou Daily, p. 001.

[3] Okinda, C., Nyalala, I., Korohou, T., Okinda, C., Wang, J., Achieng, T., Wamalwa, P., Mang, T., & Shen, M. (2020). A review on computer vision systems in monitoring of poultry: A welfare perspective. Artificial Intelligence in Agriculture, 4, 184-208.

[4] Xiao, D. Q., Wang, H. D., Liu, Y. F., Li, W. G., & Li, H. B. (2024). DHSW-YOLO: A duck flock daily behavior recognition model adaptable to bright and dark conditions. Computers and Electronics in Agriculture, 225, 109281.

[5] Ji, H. Y., Zhang, Z., Teng, G. H., Zhou, Z. Y., Liu, M. L., Ge, S. J., & Liu, J. (2024). Estimation of chicken cage' s feed intake based on improved YOLOv8n in stacked-cage-raising system. Transactions of the Chinese Society of Agricultural Engineering, 40(24), 218-225.

[6] Pei, W., Wang, Y. C., Cuan, K. X., Lin, W. Y., Shi, W. Q., Liu, Z. Y., & Wang, K. Y. (2025). Real-time detection method of dead chickens in cages using improved YOLOv8n. Transactions of the Chinese Society of Agricultural Engineering, 41(6), 170-178.

[7] Feng, Y. H., Li, W., Guo, Y. H., Wang, Y. F., Tang, S. J., Yuan, Y. C., & Shen, L. L. (2024). GooseDetectlion: A Fully Annotated Dataset for Lion-head Goose Detection in Smart Farms. Scientific Data, 11, 980.

[8] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 779-788). Las Vegas, NV, USA.

[9] Alif, M., & Hussain, M. (2024). YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in agriculture. Computers and Electronics in Agriculture, 225, 109281.

[10] Khanam, R., & Hussain, M. (2024). YOLOv11: An Overview of the Key Architectural Enhancements. Journal of Field Robotics, 41(2), 1-25.

[11] Jani, M., Fayyad, J., Younes, & Najjaran, H. (2023). Model Compression Methods for YOLOv5: A Review. arXiv:2307.11904v1.

[12] Cai, X. H., Lai, Q. X., Wang, Y. W.,WANG, W. G.,Sun, Z. R, & Yao, Y. Z. (2024). Poly Kernel Inception Network for Remote Sensing Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 27706-27716

[13] Chen, Z. X., He, Z. W., & Lu, Z. M. (2023). DEA-Net: Single image dehazing based on detail-enhanced convolution and content-guided attention. IEEE Transactions on Image Processing, 33, 1002 - 1015.

[14] Wang, J. W., Xu, C.,Yang, W. & Yu, L. (2022). A Normalized Gaussian Wasserstein Distance for Tiny Object Detection. arXiv:2110.13389v2.

[15] Zhang, Y. F., Ren, W. Q.,Zhang , Z.,Jia, Z.,Wang, L., & Tan, T. N. (2022). Focal and efficient IOU loss for accurate bounding box regression. Neurocomputing,506,146–157.