Methods for Ground Target Recognition from an Aerial Camera on a Helicopter Using the MISU-YOLOv8 Model in Dark and Foggy Environments

Houbin Wang[1,2,], Yongwei Wang[3], Junyi Liu[4], Jianing Chang[1], Huanran Shu[1], Kaidi Sun[1]*

1 School of Resources and Environmental Engineering, Ludong University, Yantai, Shandong Province, China

2 Ruiluweijun (Yantai) Information Technology Co., Ltd., Yantai, Shandong, China

3 Wenchang county Yuanzhuang town people's government, Jining, Shandong Province, China

4 School of Mechanical Engineering, Liaoning Technical University, Fuxin, Liaoning Province, China

ABSTRACT

Helicopters are critical aerial platforms, and their operational capability in complex environments is crucial. However, their performance in dark and foggy conditions is limited, particularly in ground target recognition using onboard cameras due to poor visibility and lighting conditions. To address this issue, we propose a YOLOv8-based model enhanced to improve ground target recognition in dark and foggy environments. The MS block is a multi-scale feature fusion module that enhances generalization by extracting features at different scales. The improved Residual Mobile Block (iRMB) incorporates attention mechanisms to enhance feature representation. SCINet, a spatial-channel attention-based network, adaptively adjusts feature map weights to improve robustness. UnfogNet, a defogging algorithm, enhances image clarity by removing fog. This integrated approach significantly improves ground target recognition capabilities. Unlike traditional models, AOD-Net generates clean images via a lightweight CNN, making it easily integrable into other deep models. Our MISU-YOLOv8 model outperforms recent state-of-the-art real-time object detectors, including YOLOv7 and YOLOv8, with fewer parameters and FLOPs, improving YOLOv8's Average Precision (AP) from 37% to over 41%. This work can also serve as a plug-and-play module for other YOLO models, this advancement provides robust technical support for helicopter reconnaissance missions in complex environments.

Keywords: YOLOv8; Multi-Scale Block; Inverted Residual Mobile; Self-Calibrated Illumination; Dehazing; Object Detection

1 INTRODUCTION

With the continuous development of modern warfare and civilian applications, helicopters are increasingly tasked with operations and rescue missions in complex environments. Dark and foggy conditions pose higher demands on the navigation and target recognition capabilities of helicopters. When helicopters perform missions at night or under low visibility conditions, their onboard cameras face significant challenges. Insufficient lighting results in decreased image quality, making it difficult to extract target features [1]. The increased noise and interference in dark environments further complicate target recognition. Additionally, the dynamic changes during helicopter flight demand higher stability and accuracy in target recognition.

The resolution of issues such as the determination and prevention of traffic accidents at night, the prevention of forest fires captured by surveillance cameras, nighttime facial recognition for theft prevention, and nighttime reconnaissance along national borders all rely on low-light image processing technology [2]. On the other hand, high-level visual tasks (such as target detection and semantic segmentation) that benefit from improved image quality are also of significant research value [3]. However, in the process of acquiring low-light images, factors such as insufficient ambient light, obstructions, and equipment limitations often result in low-quality images characterized by insufficient brightness, blurriness, and excessive noise. To obtain clear and informative images, Chen et al. processed the acquisition equipment and extended the exposure time to achieve better results, but this inevitably caused blurring due to jitter and object movement [4]. Increasing the sensitivity directly can enhance image brightness, but it also increases noise. Therefore, the study of low-light image enhancement algorithms at the image processing level has become a topic of great significance.

2 RELEATED WORK

Aerial recognition employs an overhead perspective, with varying flight altitudes and constantly changing target directions [5]. The captured image ranges vary in size; the higher the flight altitude, the higher the content of small ground targets, making recognition more challenging. Images taken from the ground and general recognition algorithms are not suitable for this purpose. Concurrently, with the continuous development of deep learning, the recognition accuracy of deep learning algorithm network models is continuously improving [6]. However, this is accompanied by increasing network complexity, higher computational demands, and larger model weight files, making complex network models difficult to deploy in practice. Therefore, it is necessary to lighten the recognition model for easy transplantation to embedded devices [7].

In response to such issues, many scholars have conducted research. For example, Qiao Mengyu et al. (2020) inserted the ELU function as an activation function into the lightweight MobileNet network, and the improved algorithm surpassed mainstream lightweight target detection algorithms in both detection accuracy and recognition speed of military targets on the battlefield [8]. Liu Kang et al. improved the Yolov5 by incorporating a channel-spatial attention mechanism to enhance target feature extraction capability and adopting the α-CIoU loss function as the bounding box loss function, resulting in a 6.4% accuracy improvement [9].

Qiu Hao et al. introduced a lightweight channel attention mechanism into Yolov5n to enhance the extraction of effective information from feature maps, added an adaptive spatial feature fusion module, and used the EIoU loss function to accelerate convergence and improve detection accuracy, achieving a 6.1% accuracy improvement [10]. Niu Weihua et al. integrated a small target detection layer into the aggregation network structure of Yolov7 and incorporated a channel-spatial attention mechanism in the backbone network, introducing the SioU Loss localization loss function, resulting in a 2.8% accuracy improvement [11]. Fu Jinyi et al. replaced the convolution modules of the head and neck with a partial convolution of channel features, embedded CAM to enhance the perception of deep feature details, and achieved an 8.7% improvement in small target detection accuracy [12].

YOLOv8 is the latest version in the YOLO series, distinguished by its high efficiency, accuracy, and minimal model memory usage. Based on YOLOv8, we propose a dark and foggy target recognition model called MISU-YOLOv8, tailored for challenging helicopter detection in complex environments. The main contributions are as follows:

Dataset Construction: We compiled a ground object detection dataset covering 20 types of common ground targets. Strict screening ensured that only high-quality photos were included, followed by manual annotation to accurately depict the ground targets.

Innovative Technologies: Our method integrates several advanced techniques: multi-scale block, inverted residual mobile, self-calibrated illumination, and dehazing. These technologies effectively reduce model parameters, mitigate overfitting, and lower computational and memory requirements. Additionally, our method significantly enhances the model's accuracy in recognizing ground targets.

MISU-YOLOv8 Method: We introduce the MISU-YOLOv8 method, which is lighter and more accurate than the original YOLOv8 method. It features fewer parameters, fewer FLOPS, higher FPS, and easier deployment, thus delivering superior performance.

3 MODEL ESTABLISHMENT AND SOLUTION

3.1 Model Structure of the YOLOv8s Network

YOLOv8, building on YOLOv5, incorporates both BoS (Bag of Specials) and BoF (Bag of Freebies) strategies for improvements. First, the CSP module in the backbone network has been modified. The C2f module combines contextual information and high-level features through cross-stage partial bottlenecks and two convolution operations, enhancing detection performance. The BoF strategy's DFL (Distribution Focal Loss) can handle continuous labels, optimizing both classification and bounding box regression, which further improves the detection performance for small targets. A standout feature of YOLOv8 is its modular design, making it easy to extend and modify, thus improving the model's scalability. It supports multiple export formats and can run on both CPUs and GPUs, offering efficient model deployment capabilities. However, a drawback is the continued use of the FPN-like neck, which can result in information loss during cross-layer interactions.

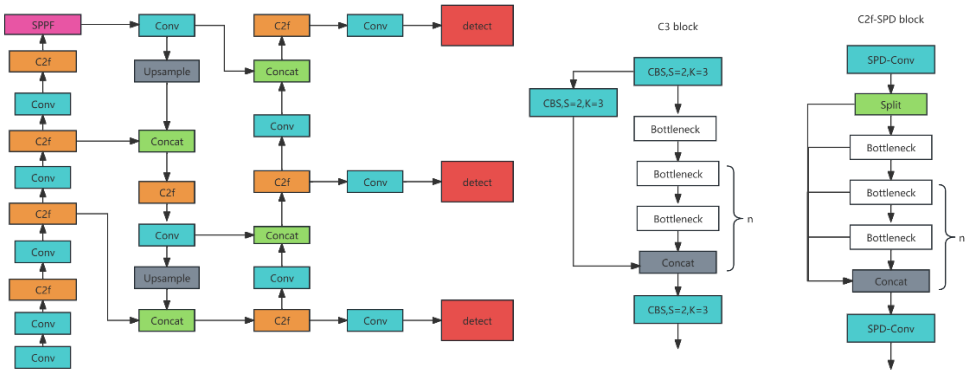

Fig. 1: YOLO v8 model structure

3.2 Multi-Scale Building Block Design

Based on previous analysis, we introduce an innovative block named the MS-Block, incorporating a hierarchical feature fusion strategy to improve the performance of real-time object detectors in capturing multi-scale features while ensuring rapid inference speed. Let $X \in RH \times W \times C$ represent the input feature. After undergoing a 1×1 convolution, the channel dimension of X is expanded to $n \times C$. Subsequently, X is divided into n distinct groups, denoted as X_i where $i \in 1, 2, 3, \ldots, n$. To reduce computational expense, we set n to 3. It's important to note that aside from $X1$, each group is processed through an inverted bottleneck layer, indicated by $IBk \times k()$ where k denotes the kernel size, to produce Yi. The mathematical expression for Yi is as follows:

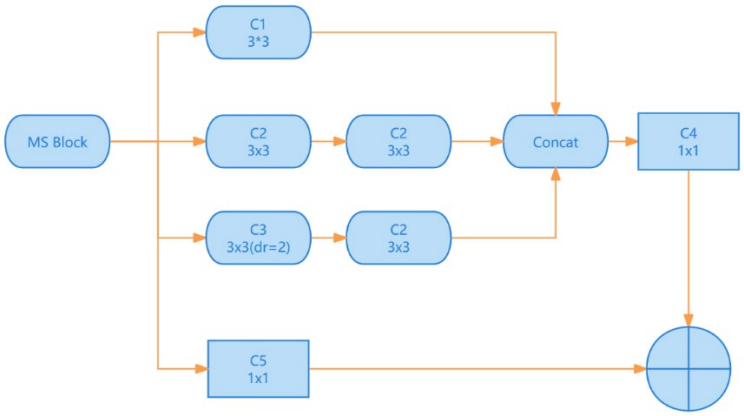

Fig. 2: Structure diagram of mult-scale building block

The MS_block takes the feature maps output by the attention prediction sub-network as the input for the sub-model. It then undergoes a 3×3 convolution kernel, two 3×3 stacked convolution blocks (equivalent to a 5×5 receptive field), and a combination of ordinary 3×3 and 3×3 dilated convolutions (dilation rate = 2, equivalent to a 7×7 receptive field). After three layers of convolutional processing, the features are fused and then compressed using a 1×1 convolution kernel to reduce the number of channels. Finally, the features are merged with those processed only by a 1×1 convolution and output to the next module. Different receptive fields can extract different targets, and multi-receptive field extraction is beneficial for

discovering more defect features.

This network is improved based on the Feature Pyramid Network (FPN) and Path Aggregation Network (PANet) to facilitate effective interaction and information transfer between features at different scales. FPN constructs a feature pyramid in a top-down manner, integrating feature maps of different resolutions and semantic levels, thereby enhancing the expression capabilities at each level. A content-aware feature reorganization module is introduced into the FPN, which dynamically generates adaptive upsampling kernels based on the input features, reorganizing and pooling the input features to improve the semantic quality and multi-scale adaptability of the upsampled feature maps. Specifically, BMFFN first reduces the dimensionality of the channels for the three scales C3, C4, C5 output by the backbone network. Then, each scale is added to the result of the upsampling of the next lower layer in a top-down manner to obtain the fused feature maps of the three scales P3, P4, P5. This enriches the semantic information at each scale while maintaining high spatial resolution.

PANet, or the Feature Pyramid Attention Network, performs downsampling and fusion operations on feature maps of different scales in a bottom-up manner. It employs a max-pooling module based on modulated deformable convolution to enhance the adaptability and expression capability of the features, thereby improving the detail information and complementarity of the features. Specifically, PANet first increases the dimensionality of the channels for the three scales of fused feature maps P3, P4, P5 output by the FPN. Then, starting from the smallest scale, each scale's feature map is added to the downsampled feature map of the next layer to obtain the fused feature maps of three scales N3, N4, N5. This enhances the connections between different scales. By combining FPN and PANet, a bidirectional multi-scale feature fusion is achieved, making full use of feature information at different scales, thereby improving the performance of object detection.

In the MS-Block, the hierarchical feature fusion strategy plays a crucial role. This strategy allows the MS-Block to effectively handle multi-scale features, which is vital for the accurate detection of objects of varying sizes in real-time applications. The design ensures that the process remains computationally efficient, which is critical for maintaining the speed required in real-time systems. By increasing the channel dimension via a 1×1 convolution and then splitting the input into distinct groups, the MS-Block leverages the inverted bottleneck layers to refine each group except for the first one. This approach balances the need for detailed feature extraction with the necessity of keeping the computational load manageable.

$$Y_i = \begin{cases} X_i, & i = 1 \\ IB_{k\times k}(Y_{i-1} + X_i) & i > 1 \end{cases} \quad (1)$$

In accordance with the formula, we avoid connecting the inverted bottleneck layer to $X1$. This approach enables $X1$ to function as a cross-stage connection, thereby retaining information from previous layers. This preservation of earlier information is crucial for maintaining the integrity and continuity of the feature extraction process. Once all splits are processed, we concatenate them and apply a 1×1 convolution. This step is essential for allowing interaction among the splits, each of which encodes features at different scales. The 1×1 convolution not only merges these multi-scale features but also standardizes the channel numbers, which is particularly important as the network architecture becomes more complex and deeper.

3.2 Inverted Residual Mobile Block

The inverted residual structure first performs a 1x1 convolution for dimension expansion on the residual module, then carries out depthwise convolution, and finally uses projection convolution for dimension reduction. This structure initially uses an expansion factor a (with a set to 2 in this paper) to expand the input feature map's channels by C times a, thereby obtaining rich shallow features. Following this, a 1x1 convolution is applied for dimension expansion, which allows the acquisition of graphical features after expansion. The depthwise convolution structure splits the single step of standard convolution into two steps: first, depthwise convolution is performed using M convolution kernels of size D_K times D_K and depth 1, and then pointwise convolution is performed using N convolution kernels of size 1x1 and depth M. The depthwise convolution is responsible for filtering, while the pointwise convolution is responsible for channel transformation. The standard convolution parameters are C_1, and the depthwise convolution parameters are C_2.

$$\frac{C_2}{C_1} = \frac{D_K^2 M D_F^2 N}{D_K^2 M D_F^2 + M N D_F^2} = \frac{1}{N} + \frac{1}{D_K^2} \tag{2}$$

Compared to standard convolution, depthwise convolution reduces the number of model parameters, thereby improving the model's real-time detection capability. Finally, during the dimensionality reduction operation, the Linear activation function is used instead of the ReLu6 activation function, effectively reducing the information loss caused by the nonlinear activation function. Based on the inductive Meta-Mobile Block, we present an advanced and efficient modern Inverted Residual Mobile Block (iRMB) from a detailed perspective.

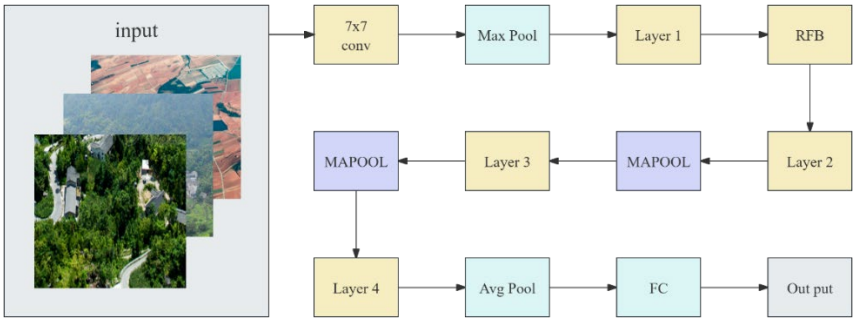

Fig. 3: Paradigm of iRMB

The function F in iRMB is modeled as a combination of Multi-Head Self-Attention (MHSA) and Convolution operations, formulated as $F() = Conv(MHSA())$.

Table 1: A toy experiment for assessing iRMB.

Model	Params↓	FLOPs↓	Top-1↑
DeiT-Tiny	5.7M	1258	72.2
DeiT-Tiny w/iRMB	4.9M-14%↓	1102 -156M↓	74.3 +2.1%↑
PVT-Tiny	13.2M	1943	75.1
PVT-Tiny w/iRMB	11.7M -11%↓	1845 -98M↓	75.4 +0.3%↑

Following the criteria outlined, the function F in iRMB is modeled as a combination of Multi-Head Self-Attention (MHSA) and Convolution operations, formulated as $F() = Conv(MHSA())$. This design integrates the efficiency of Convolutional Neural Networks

(CNNs) for local feature modeling with the dynamic capability of Transformers for learning long-distance interactions. However, a straightforward implementation can be prohibitively expensive for two main reasons, first, The factor λ is typically greater than one, meaning the intermediate dimension is a multiple of the input dimension. This leads to a quadratic increase in parameters and computations with respect to λ. Therefore, the components of F should either be independent of or linearly dependent on the number of channels. secondly, The floating point operations (FLOPs) for MHSA scale quadratically with the total number of image pixels, making the cost of a naive Transformer implementation prohibitive.

To balance model cost and accuracy, we employ efficient Window-MHSA (W-MHSA) and Depth-Wise Convolution (DW-Conv) with a skip connection. The parameters and FLOPs required for obtaining queries Q and keys K in W-MHSA are quadratic with respect to the channel dimension. To mitigate this, we use the unexpanded input X to calculate the attention matrix more efficiently, i.e., $Q = K = X(X \in R\hat{}C \times H \times W)$, while the expanded value X_e serves as the value matrix $V(V \in R\hat{}\lambda C \times H \times W)$. This improvement, termed Expanded Window MHSA (EW-MHSA), is more practical and is formulated as:

$$\mathcal{F}(\cdot) = (DW - Conv, Skip)\big(EW - MHSA(\cdot)\big). \tag{3}$$

This passage explains an efficient implementation strategy for a particular type of neural network operation, specifically within the context of Multi-Head Self-Attention (MHSA). MHSA is typically used in scenarios where the number of channels (features) remains consistent $(\lambda = 1)$. When the channels expand $(\lambda > 1)$, the number of floating points operations (FLOPs) required for multiplying the attention matrix by the expanded input (X_e) increases significantly by a factor of $\lambda - 1$. However, the transformation from the original input (X) to the expanded input (V) involves only linear operations, such as those performed by a Multi-Layer Perceptron (MLPe). The key proposition is that when the number of groups in the MLPe matches the number of heads in the weighted MHSA (W-MHSA), the result of their operations remains unchanged even if the order of operations is swapped. This means you can perform matrix multiplication before applying the MLPe to reduce the overall computational cost (FLOPs). By default, the implementation uses matrix multiplication before applying the MLPe to achieve this efficiency.

3.3 Self-calibration Illumination

This paper introduces the Self-Calibrated Illumination (SCI) algorithm to enhance coal flow foreign object images on belt conveyors. The SCI algorithm is an unsupervised learning algorithm based on Retinex theory. By constructing a progressive illumination optimization, it establishes a cascading illumination learning process with weight sharing and introduces a self-calibration module. This defines an unsupervised training loss to achieve rapid, flexible, and robust image enhancement in low-light scenarios. According to the previously introduced Retinex image decomposition theory, an image can be considered as the product of the object reflection image and the illumination image. The object reflection image represents the inherent properties of the object, unaffected by illumination, and by estimating the illumination image, the reflection image can be calculated to enhance the image. Similarly, a low-light image can be considered as the product of a clear image and an illumination image.

Among $Y = Z \otimes X$, Y represents the low-light image; Z represents the clear image unaffected by illumination, which is the enhanced image; X represents the illumination image, which is the main enhancement modification in our method; In previous research, it was common to estimate the illumination image to eliminate the estimated illumination image to obtain a clear image unaffected by illumination, thereby enhancing the low-light image. The self-calibrated illumination learning algorithm was inspired by the multi-stage illumination optimization process in deep learning, constructing a progressive illumination optimization process.

$$F(X^t): \begin{cases} X^0 = Y \\ u^t = H_\theta(X^t) \\ X^{t+1} = X^t + u^t \end{cases} \tag{4}$$

Among them, X_t and $U_t$represent the illumination image and residual term at stage t respectively. The residual term U_t is the update parameter of the progressively updated illumination image X_t, which also represents the reflection relationship between the illumination image X and the low-light image Y; H_θ represents the illumination image estimation network, which learns the residual term through U_t parameterized calculation θ. It is noteworthy that at each stage of learning the illumination image, the framework and weights of the illumination image estimation network are the same. Currently, most scholars' research on low-light image enhancement is based on the theory that there is a linear relationship between normal-light and low-light images. Illumination image learning with weight sharing learns the residuals to map between normal-light and low-light images, thereby reducing computational complexity while ensuring performance and stability. However, since the aforementioned method constructs a progressive illumination image optimization process, the multiple weight-sharing stages inevitably increase inference costs. Therefore, it is necessary to design a calibration module so that the results of each stage converge to the same value. This way, during the testing phase, only the results of the first stage are needed for image enhancement, thereby reducing computational costs.

$$G(X^t): \begin{cases} Z^t = Y \otimes X^t, \\ S^t = K_\vartheta(Z^t), \\ V^t = Y + S^t \end{cases} \tag{5}$$

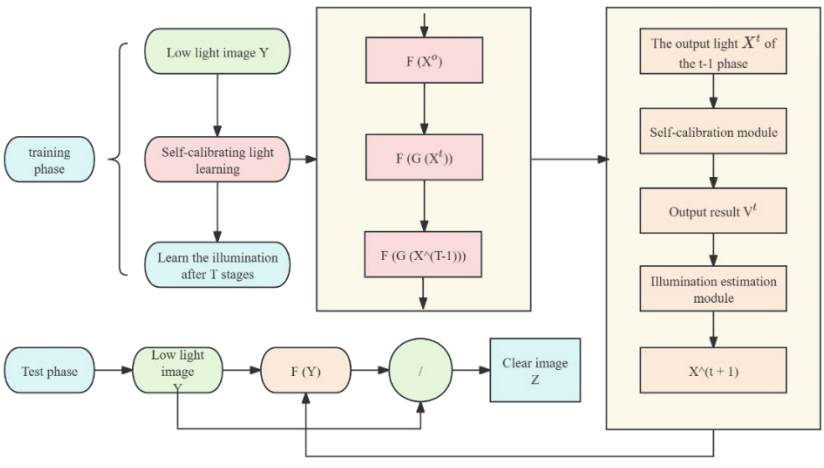

Fig. 4: Self-calibrating illumination learning process

The previously introduced weight-sharing illumination image learning takes the input of each stage from the output of the previous stage (except for the first stage), with the initial input being the low-light image. The self-calibration module connects the low-light image with the input of each stage, enabling the exploration of convergence behavior between stages and ultimately achieving the goal of balancing the convergence results of each stage. a schematic diagram is presented in Figure 4.

3.4 Unfog Network base on AOD-Net

Riding the wave of deep learning, dehazing algorithms based on deep learning have also made significant progress. Compared to traditional methods, deep learning algorithms generally achieve better dehazing effects and are often end-to-end approaches. Examples include DehazeNet, which combines dark channel, color attenuation prior, and maximum contrast, or the Gated Context Aggregation Network (GCANet) that employs adversarial learning. Each of these deep learning algorithms has its own advantages. However, since this paper focuses on vehicle and pedestrian detection in foggy weather, there are higher requirements for the real-time performance and lightweight nature of the dehazing algorithm. Therefore, this paper ultimately adopts the smaller and more real-time AOD-Net. AOD-Net is a lightweight dehazing network based on deep learning. This algorithm also relies on the atmospheric dehazing model, but the difference is that AOD-Net merges the atmospheric transmission $t(x)$ and atmospheric light value A into a single $K(x)$.

$$K(x) = \frac{\frac{1}{t(x)}(I(x) - A) + (A - b)}{I(x) - 1} \tag{6}$$

b is a constant bias with a value of 1. AOD-Net integrates the atmospheric light value and the atmospheric transmission rate, simplifying the conversion relationship between foggy and non-foggy images and reducing error. Now, by simply knowing the value of $K(x)$, a clear image can be generated. The central idea of AOD-Net is thus revealed: by establishing an adaptive deep model, it uses a convolutional neural network to estimate the K value from the input foggy images. Once the K value is obtained, a new fog-free image can be generated through the clear image generation module according to the transformed formula.

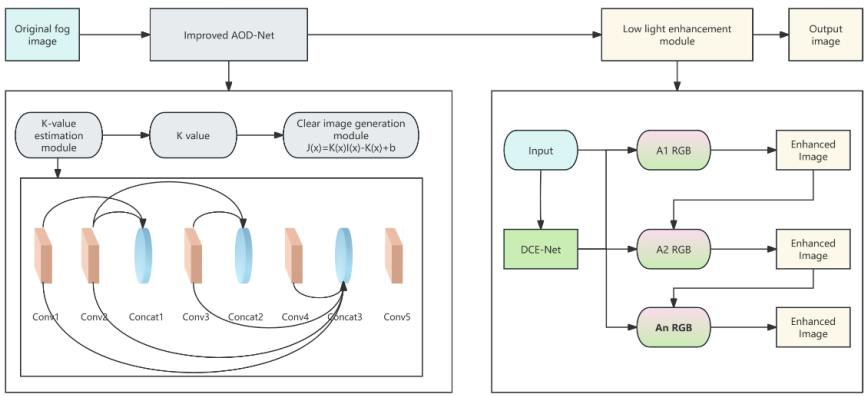

Fig. 5: The structure of the AOD-Net model

This paper introduces dynamic convolution to replace the standard convolution in the AOD-Net network. Based on the input attention mechanism, dynamic convolution aggregates multiple parallel convolution kernels. Since dynamic convolution uses smaller convolution kernels, it achieves higher computational efficiency. Additionally, because these convolution kernels are aggregated in a nonlinear manner, dynamic convolution has a stronger feature representation capability. $y_z = g(W_z^{\pi} + b)$, W and b are the weight matrix and bias vector, respectively, and g is the activation function.

$$y_d = g\left(\widetilde{W}^T(x)x + \tilde{b}(x)\right) \tag{7}$$

$$\widetilde{W}(x) = \sum_{k=1}^{\kappa} \pi_k(x)\widetilde{W}_k \tag{8}$$

$$\tilde{b}(x) = \sum_{k=1}^{\kappa} \pi_k(x)\tilde{b}_k \tag{9}$$

π is the weight vector of the k th linear function, and this weight vector varies with x. The dynamic perceptron is a combination of linear models of the input, thus possessing stronger fitting capabilities.

The ECA (Efficient Channel Attention) module is an improved attention module based on SENet. This module adds only a small number of parameters while significantly enhancing performance. its computational process is as follows:

Calculate the average value of each channel

$$M_c = \frac{1}{H \times W}\sum_{i=1}^{h}\sum_{j=1}^{W} X_{cij} \tag{10}$$

Here, X is the input feature map, C is the channel index representing a specific channel in the feature map, and i and j are spatial dimension indices representing a specific position in the feature map, where i indicates the height dimension and j indicates the width dimension. Apply a learnable scaling factor.

$$M_c' = \theta(M_c) = W_\theta \cdot M_c \tag{11}$$

Here, W_θ is a learnable weight parameter matrix used to adjust the average value.

$$A_c = \sigma(M_c') = \frac{1}{1 - exp(-M_c')} \tag{12}$$

Here, σ is the Sigmoid function, used to map the scaled average value to the attention weight within the range of (0,1). $Y_{cij} = A_c \cdot X_{cij}$, Y_{cij} is the feature value with the applied attention weight. Finally, the output feature map weighted by the ECA attention is obtained.

AOD-Net uses a single mean squared error (MSE) loss function to measure the squared difference between the predicted and true values, which can easily lead to local optima. This paper adopts the MS-SSIM-L2 loss function to replace the original MSE loss function to enhance the dehazing performance of the entire network. First, the SSIM value for each scale is calculated, and then they are weighted and averaged to obtain the MS-SSIM value. The scale

index s usually ranges from 1 to n.

$$SSIM_s(G,T) = \frac{2\mu_G\mu_T + c1}{\mu_G^2 + \mu_G^2 + c1} \cdot \frac{2\sigma_{GT} + c2}{\sigma_G^2 + \sigma_T^2 + c2} \tag{13}$$

$$MS - SSIM(G,T) = \frac{1}{N}\sum_{s=1}^{N} SSIM_s(G,T) \tag{14}$$

Here, μ_G and μ_T are the means of the generated image and the target image, respectively, μ_G and μ_T are their standard deviations, μ_{GT} is their covariance, $c1$ and $c2$ are stability constants, and N is the number of scales.

$$L2(G,T) = \frac{1}{C \times H \times W}\sum_{c=1}^{C}\sum_{i=1}^{H}\sum_{j=1}^{w}(G_{cij} - T_{cij})^2 \tag{15}$$

The L2 loss measures the pixel-level differences between the generated image and the target image.

$$MS - SSIM - L2 = \alpha \cdot MS - SSIM(G,T) + \beta \cdot L2(G,T) \tag{16}$$

The final MS-SSIM-L2 loss combines the MS-SSIM and L2 losses in a weighted manner. Here, α and β are weight coefficients used to balance the relative influence of the MS-SSIM and L2 losses. Compared to the previous single mean squared error loss function, the composite loss function proposed in this section improves both brightness and contrast, making it the chosen final objective loss function for the algorithm in this paper.

4 EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Data sources

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

Due to the limited availability of datasets featuring ground objects from a helicopter perspective, the following types of ground target datasets were found online: UC Merced Land-Use Data Set, DOTA, and TGRS-HRRSD-Dataset. These three datasets all contain number of ground object photos; however, because the DOTA and TGRS-HRRSD-Dataset datasets have fewer categories for ground object classification, the dataset established by the University of California, Merced, was selected for this paper. The UC Merced Land-Use Dataset includes a total of 20 categories of scene images, each category has 100 images, totaling 2,000 images.

4.2 Data preprocessing

To enhance the robustness of the neural network, reduce model dependency, and prevent overfitting due to a small training dataset, the UC Merced Land-Use dataset was subjected to data augmentation. The data augmentation methods include the following four types: single-sample data augmentation, multi-sample data augmentation, generating new data, and learning augmentation strategies. Since the UC Merced Land-Use dataset requires geometric

operations and color transformations on the images, the data augmentation method adopted in this paper is single-sample data augmentation. By applying 11 methods, each image in the original dataset was expanded to 12 times its original size.

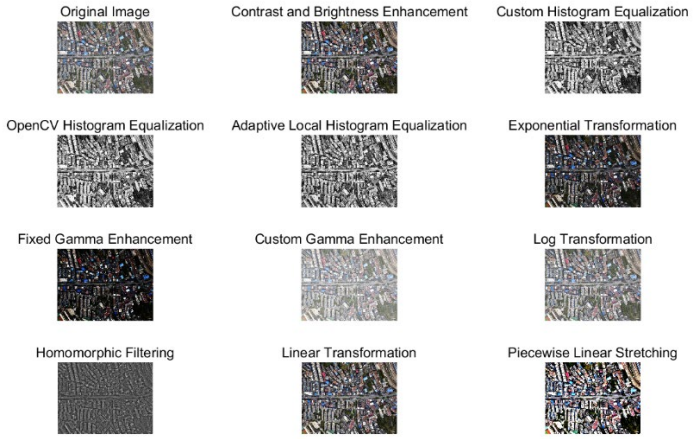

Fig. 6: Image processing effect

4.3 Experimental Environment

This experiment is based on the Windows system. The specific experimental environment, related configuration information, and initial parameter information are shown in Tables 3 and 4.

Table 3: Experimental Environment and Configuration

Name	Specification
Operating System	Windows 11
CPU	Intel(R) Core(TM) i9-10920X CPU @ 3.50 GHz, 32 GB RAM
GPU	Nvidia RTX 3090 ×2
Framework	TensorFlow 2.0
Environment Config	Python 3.6, scipy 1.5.4, keras 2.3.1, matplotlib 3.3.3, numpy 1.9.5, pandas 1.1.5, pillow 8.4.0
Name	Specification

Table 4: Initial Parameter Settings

Parameter	Value
lr	0.0006
betas	0.89
betas2	0.98
batch_size	32
image_size	64×64
epoch	200

To verify the superiority of ground target recognition based on the MISU-YOLOv8 method, training data from five different network models were selected for comparison. Under the same training parameter settings, the training and validation were conducted for these five network models. After smoothing the training accuracy curves, the results are shown in Figure 7.

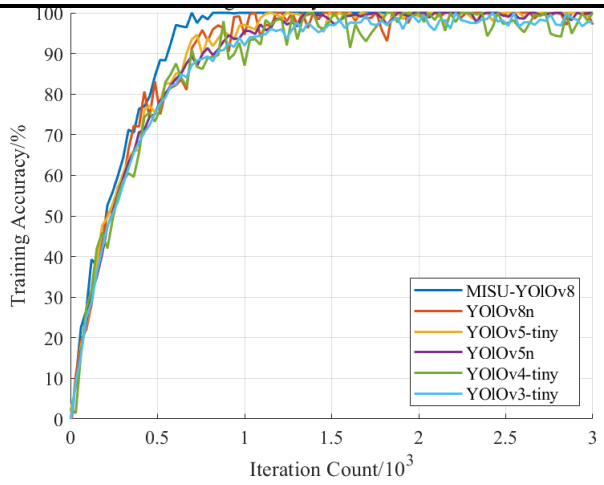

Fig. 7: Traning Accuracy Curves for 6 Models

In Figure 7, the YOLOv8 network model refined through transfer learning is denoted as MISU-YOLOv8. The figure illustrates that, with increasing iterations, the training accuracy of all five network models consistently improves and gradually stabilizes. Except for the YOLOv3-tiny network model, the training accuracy of the remaining four network models exceeds 93%. Among these models, MISU-YOLOv8 exhibits the fastest convergence rate, while YOLOv3-tiny is the slowest. After approximately 600 iterations, MISU-YOLOv8 achieves a 90% training accuracy and quickly converges to 100% after 900 iterations. Conversely, the SqueezeNet model reaches 98% training accuracy only after 1500 iterations. The training accuracy curves of the other four models are similar, with final training results ranging between 94% and 99%. The training outcomes demonstrate that the MISU-YOLOv8 network, refined through transfer learning, surpasses the other five networks in both convergence speed and training accuracy.

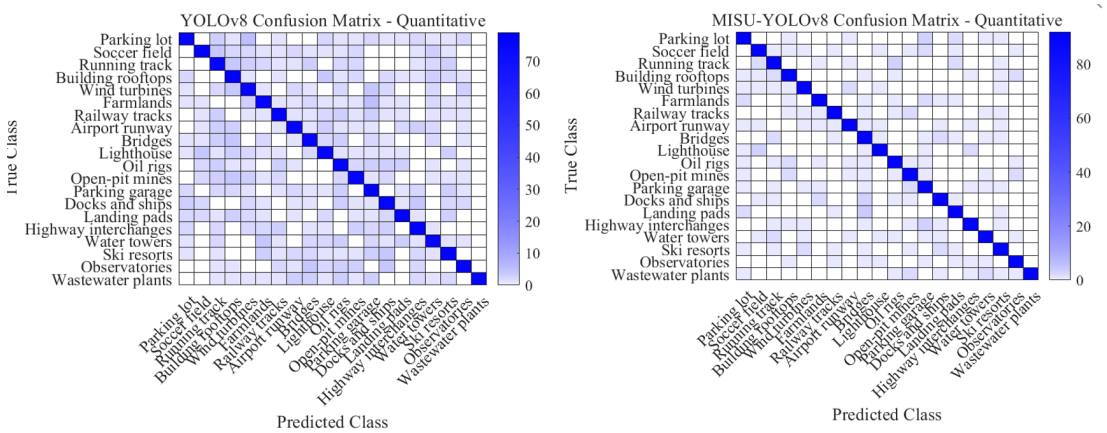

Fig. 8: confusion matrix quantiative (a) confusion matrix quantiative of YOLOv8n; (b) confusion matrix quantiative of MISU-YOLOv8

The test results indicate that the overall recognition rate of the MISU-YOLOv8 network model can reach 91.68%, while the overall recognition rate of the YOLOv8 network model is only 75.89%. The MISU-YOLOv8 network model has an average prediction accuracy of over 80% for 20 types of targets, whereas the YOLOv8 network model only has three types of targets with a prediction accuracy of over 80%. In summary, the MISU-YOLOv8 network model performs excellently in recognizing ground targets, achieving very high accuracy in ground target recognition.

Classify the distribution of the extracted bounding boxes of image data. Using the image center as the origin, measure the coordinates of the center points of each edge of the bounding boxes in image recognition. Perform two-dimensional and three-dimensional statistics, with the results shown in Figure 9 and Figure 10.

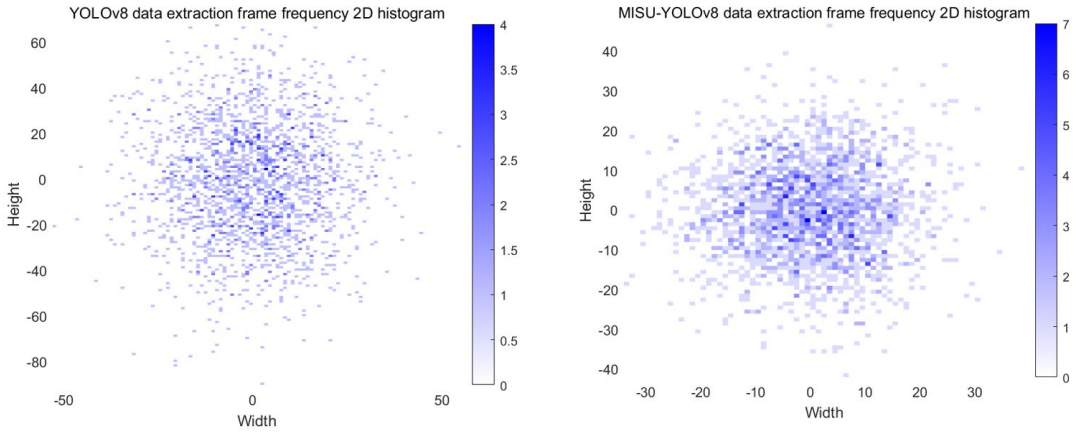

Fig. 9: data extraction frame frequency 2D histogram (a) ata extraction frame frequency 2D histogram of YOLOv8n; (b) ata extraction frame frequency 2D histogram of MISU-YOLOv8

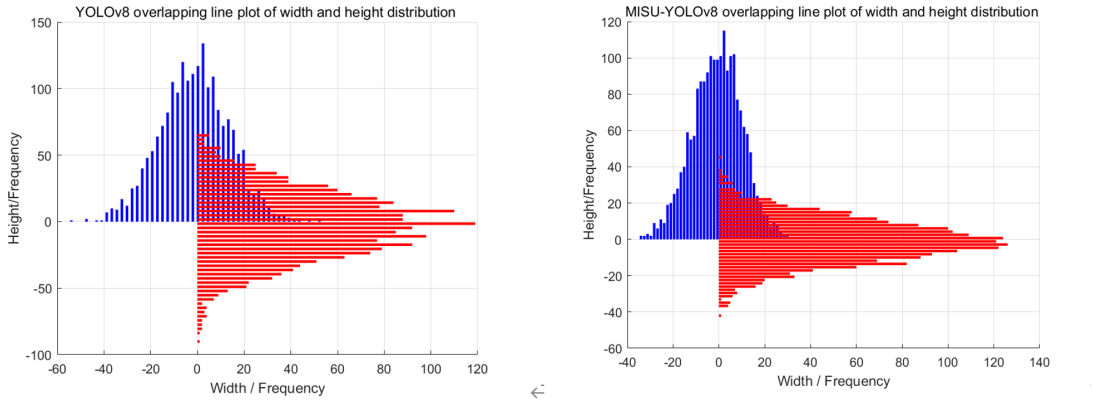

Fig. 10: Overlapping line plot of width and height distribution (a) Overlapping line plot of width and height distribution of YOLOv8n; (b) Overlapping line plot of width and height distribution of MISU-YOLOv8

Compared to YOLOv8, MISU-YOLOv8 shows significant improvements. The distribution of bounding boxes is more concentrated, exhibiting better fitting uniformity in all directions and closer to a normal distribution. MISU-YOLOv8 has 127 and 116 bounding boxes at the most densely distributed positions for width and height, respectively. The peak differences in width and height are small, indicating high data fitting accuracy.

Table 5: Comparison of results of different models

Simulations	P/%	R/%	mAP/%	F1/%	FPS
YOLOv3-tiny	79.74	80.17	81.83	78.86	561
YOLOv4-tiny	80.78	75.14	82.28	78.81	228
YOLOv5n	79.63	75.18	82.13	76.81	322
YOLOv7-tiny	82.03	73.76	81.68	76.40	351
YOLOv8n	78.93	75.40	80.08	77.68	521
MISU-YOLOv8	80.97	80.91	88.17	80.04	587

The training of the model, compared with the proposed MISU-YOLOv8 ground target

recognition algorithm and related algorithms, demonstrates the advantages of the MISU-YOLOv8 presented in this chapter. Table 5 compares YOLO with MISU-YOLOv8 and related lightweight algorithm models, where MISU-YOLOv8 shows obvious superiority in several key metrics. MISU-YOLOv8 achieves a precision of 80.97%, which is slightly lower than YOLOv7-tiny (82.03%), but stands out among other models, indicating its high accuracy in predicting targets. The recall rate of MISU-YOLOv8 is 80.91%, significantly higher than the highest value of other models at 80.17% (YOLOv3-tiny), meaning it can detect more actual targets. MISU-YOLOv8's mean average precision reaches 88.17%, the highest among all models, approximately 7.16 percentage points higher than the second-best YOLOv4-tiny, showcasing its stronger overall performance. MISU-YOLOv8 maintains a high frames per second (FPS) rate at 587, although slightly lower than YOLOv3-tiny (561 FPS) and YOLOv8 (521 FPS), yet, combined with its excellent accuracy and recall rate, indicates its strong real-time processing capability in practical applications.

Table 6: Comparison of the results of different module combinations

Models	Based Models	MS Block	iRMB	SCINet	P/%	R/%	F1/%	mAP/%	FLOPs (G)	FPS
Model1	YOLOv8				77.43	77.79	75.79	83.76	28.43	534
Model2	YOLOv8			✓	81.64	73.45	78.35	80.37	27.96	873
Model3	YOLOv8		✓		77.76	74.31	78.63	81.11	26.79	644
Model4	YOLOv8	✓			84.46	83.91	85.15	81.89	14.07	475
Model5	YOLOv8		✓	✓	81.64	75.19	75.56	81.99	28.16	675
Model6	YOLOv8	✓	✓		80.91	80.57	77.25	85.34	13.43	541
Model7	YOLOv8	✓		✓	80.47	81.15	82.77	79.69	13.12	525
Model8	YOLOv8	✓	✓	✓	80.97	80.91	80.04	88.17	12.98	587

The MS Block significantly improves precision, recall, and F1 score, while effectively reducing computational load (FLOPs). SCINet mainly enhances FPS significantly, with some improvement in precision and F1 score. iRMB contributes to improving FPS and certain performance metrics (such as F1 score). Overall, Model8, which incorporates all components, performs the best, showing significant improvements across all performance metrics. In particular, it achieves an average precision of 88.17%, indicating its strong overall performance in object detection tasks.

5 CONCLUSION

Compared to YOLOv8, MISU-YOLOv8 shows significant improvements. MISU-YOLOv8 has 127 and 116 bounding boxes at the most densely distributed positions for width and height, respectively, achieving a test recognition rate of 91.68% versus YOLOv8's 75.89%. MISU-YOLOv8 has over 80% prediction accuracy for 20 types of targets, while YOLOv8 achieves this for only three types. MISU-YOLOv8 reaches 90% training accuracy after approximately 600 iterations and converges to 100% after 900 iterations. In comparison, SqueezeNet reaches 98% training accuracy only after 1500 iterations, with other models achieving 94%-99%.

The proposed YOLO v8 model, enhanced with MS block, iRMB, SCINet, and UnfogNet, shows significant advancements in ground target recognition for helicopter-mounted vision systems, particularly in dark and foggy environments. The integration of multi-scale features

and attention mechanisms has been pivotal in improving the accuracy and robustness of target detection.

The proposed model's superior performance in complex environments not only boosts the operational capabilities of helicopters but also has broader implications. For instance, in search and rescue missions, the ability to accurately detect and recognize targets in low visibility conditions can save lives. Similarly, for inspection tasks, enhanced detection capabilities can lead to more efficient and thorough assessments, ensuring better maintenance and safety standards.

6 DATA SOURCES

The article includes some data to support the results of this research. The dataset for this article is available at https://drive.google.com/drive/folders/1UdlgHk49iu6WpcJ5467iT-UqNPpx__CC.

ACKNOWLEDGEMENTS

Thanks for the data support provided by National-level Innovation Program Project Fund "Research on Seedling Inspection Robot Technology Based on Multi-source Information Fusion and Deep Network" (No.: 202410451009); Jiangsu Provincial Natural Science Research General Project (No.: 20KJB530008); China Society for Smart Engineering "Research on Intelligent Internet of Things Devices and Control Program Algorithms Based on Multi-source Data Analysis" (No.: ZHGC104432); China Engineering Management Association "Comprehensive Application Research on Intelligent Robots and Intelligent Equipment Based on Big Data and Deep Learning" (No.: GMZY2174); Key Project of National Science and Information Technology Department Research Center National Science and Technology Development Research Plan (No.: KXJS71057); Key Project of National Science and Technology Support Program of Ministry of Agriculture (No.: NYF251050).

REFERENCES

[1] Yang, S., Zhou, D., Cao, J., & Guo, Y. (2023). LightingNet: An integrated learning method for low-light image enhancement. *IEEE Transactions on Computational Imaging*, *9*, 29-42.

[2] Fisa, R., Musukuma, M., Sampa, M., Musonda, P., & Young, T. (2022). Effects of interventions for preventing road traffic crashes: an overview of systematic reviews. *BMC public health*, *22*(1), 513.

[3] Jiang, D., Li, G., Tan, C., Huang, L., Sun, Y., & Kong, J. (2021). Semantic segmentation for multiscale target based on object recognition using the improved Faster-RCNN model. *Future Generation Computer Systems*, *123*, 94-104.

[4] Ren, Z., Fang, F., Yan, N., & Wu, Y. (2022). State of the art in defect detection based on machine vision. *International Journal of Precision Engineering and Manufacturing-Green Technology*, *9*(2), 661-691.

[5] Sun, N., Zhao, J., Shi, Q., Liu, C., & Liu, P. (2024). Moving target tracking by unmanned aerial vehicle: A survey and taxonomy. *IEEE Transactions on Industrial Informatics*.

[6] Tulbure, A.-A., Tulbure, A.-A., & Dulf, E.-H. (2022). A review on modern defect detection models using DCNNs–Deep convolutional neural networks. *Journal of Advanced Research*,

35, 33-48.

[7] Sharma, B. B., Raffik, R., Chaturvedi, A., Geeitha, S., Akram, P. S., Natrayan, L., Mohanavel, V., Sudhakar, M., & Sathyamurthy, R. (2022). Designing and implementing a smart transplanting framework using programmable logic controller and photoelectric sensor. *Energy Reports*, *8*, 430-444.

[8] Zhang, Z., Xie, X., Yang, M., Tian, Y., Jiang, Y., & Cui, Y. (2023). Improving social media popularity prediction with multiple post dependencies. *arXiv preprint arXiv:2307.15413*.

[9] Li, J., Zheng, C., Chen, P., Zhang, J., & Wang, B. (2025). Small object detection in UAV imagery based on channel-spatial fusion cross attention. *Signal, Image and Video Processing*, *19*(4), 302.

[10] Liu, P., Wang, Q., Zhang, H., Mi, J., & Liu, Y. (2023). A lightweight object detection algorithm for remote sensing images based on attention mechanism and YOLOv5s. *Remote Sensing*, *15*(9), 2429.

[11] Hua, W., Chen, Q., & Chen, W. (2024). A new lightweight network for efficient UAV object detection. *Scientific Reports*, *14*(1), 13288.

[12] Jinyi, F., Zijia, Z., Wei, S., & Kaixin, Z. (2024). Improved YOLOv8 Small Target Detection Algorithm in Aerial Images. *Journal of Computer Engineering & Applications*, *60*(6).