ABSTRACT Flies carry pathogens that endanger the health of humans and animals. The color and shape of the fly species are very similar, which is difficult to recognize. This paper proposes a fly species recognition method based on improved RetinaNet and convolutional block attention module (CBAM). Firstly, the proposed method used ResNeXt101 as a feature extraction network, and the improved CBAM called Stochastic-CBAM was added. Then, we built a multi-scale feature pyramid through an improved feature pyramid network (FPN) and integrated multi-level feature information. Finally, the small full convolutional network (FCN) was used as the classification subnet and the bounding box regression subnet. The Kullback-Leibler (KL) loss replaced smooth L1 loss as a bounding box regression loss function for learning bounding box regression and positioning uncertainty at the same time. We experimentally compared the proposed method with other state-of-the-art methods on the established dataset. Experimental results showed that the mean Average Precision (mAP) of this method reached 90.38%, which was better than the state-of-the-art methods. The average time to recognize a single image was 0.131s. This method has a good detection effect on the fly species recognition.

INDEX TERMS Fly species recognition, RetinaNet algorithm, attention convolution module, convolutional neural network.

I. INTRODUCTION

With the continuous development of international trade, the types and quantities of inbound goods have increased significantly, which may lead to the invasion of alien species. Therefore, the customs strengthened the inspection and quarantine of the goods to ensure the biological safety of the country. After the customs quarantine personnel intercept the vector organisms on site, they need to be sent to the laboratory for species recognition and pathogenic testing to determine whether they carry pathogens. If the quarantine personnel quickly recognize vector organisms on site, it is meaningful to determine the risk of transmission of the vector organism. Vector organisms such as flies need to be strictly monitored. Flies have the characteristics of large number, rapid propagation and wide distribution. Some fly species carry and transmit pathogens, which can lead to a variety of diseases. For example, Muscina stabulans can cause polio and foot-and-mouth disease, and Neomyia laevifrons can transmit diseases such as typhoid fever, dysentery and trachoma. Therefore, effective recognition of flies is beneficial to prevent the invasion of alien species, prevent infectious diseases and ensure the health of humans and animals.

Generally, the traditional insect recognition methods use the combination of feature extraction and classifier. Yao et al. [1] proposed a method to remove the background by using the color difference of two images with and without insects. 156 features such as the color, shape, and texture of each pest are extracted into a support vector machine (SVM) classifier with a radial basis kernel function to classify four Lepidoptera rice pests. Kang et al. [2] proposed the use of shape features to train an artificial neural network classifier, which is used to recognize butterflies. Wen et al. [3] used invariant area feature detectors and scale-invariant feature transformations to extract features, and compared the classification effect of six classifiers on five orchard insects. Wang et al. [4] designed an automatic recognition system for insect specimen images, using artificial neural networks and support vector machines to train and learn image features. However, these methods need to manually select feature...
parameters and rely on manual design features which lead to a low recognition accuracy and detection efficiency.

In recent years, with the rapid development of artificial intelligence [5] and machine learning [6], convolutional neural network (CNN) has been widely used in the field of computer vision [7]. It has achieved breakthrough success, especially in object detection. There are two kinds of deep learning [8] methods in the field of object detection, which including two-stage detection methods and one-stage detection methods. The two-stage object detection methods mainly include the R-CNN series [9], [10]. For example, Mask R-CNN [11] is proposed by He et al., which uses ResNet [12] and FPN [13] to extract features. It obtains many regions of interest on the feature map, and then sends them to the region proposal network [14] for RoIAlign operation. Finally, classification, bounding box regression and mask generation are performed. Cai et al. proposed a cascaded R-CNN [15]. It consists of a series of detectors, and sets different Intersection over Union (IoU) thresholds to train better detectors. However, the two-stage object detection methods have the problem of the slow detection speed. There are still some one-stage object detection methods [16], [17]. Redmon et al. proposed YOLOv3 [18], which uses DarkNet53 of the residual network structure. Although the detection speed of this method is fast, the accuracy of detection is not high. Zhao et al. proposed M2Det [19], which applies a multi-level feature pyramid network to construct effective feature pyramids for detecting objects of different scales. This method has good detection accuracy and detection speed, but the accuracy is low when recognizing small objects. Liu et al. proposed the RetinaNet model [20], which extracts features to build FPN by feature extraction network. It classifies and locates objects through classification subnets and regression subnets. This method has fast detection speed and high accuracy, but the accuracy of different object recognition with similar characteristics is still low. There are some models to recognize different objects with similar characteristics. Zheng et al. [21] proposed a trilinear attention sampling network (TASN) which learns rich feature representations from hundreds of part proposals. Hu et al. [22] proposed a weakly supervised data attention network (WS-DAN), which combines weakly supervised learning with data augmentation to recognize different objects with similar features. In this paper, the recognition object is flies. Different species of flies have high similarity in color, texture and shape.

Aiming at the above problems, we propose a method of fly species recognition based on improved RetinaNet and CBAM. For the purpose of improving the accuracy of similar object recognition, we improve FPN network, introduce a bottom-up path augmentation method, and add the improved CBAM [23] to the feature extraction network of the RetinaNet model. The KL loss function [24] replaces the smooth L1 loss [9] function to learn bounding box regression and positioning uncertainty. The fly species recognition model is obtained by training the network, which is used to test flies. The main contributions of this paper are as follows. (1) On the basis of FPN, we designed Cross-Feature Pyramid Level Fusion (CFPLF) structure to increase the rich semantic information of low-level features. (2) We introduced a bottom-up path augmentation [25] method to enhance the positioning capability of the entire feature level. (3) We improved CBAM by adding stochastic-pooling to obtain a more complete attention map, and better grasped the global information of the image receiving field. (4) The KL loss function replaces the smooth L1 loss function to learn bounding box regression and positioning uncertainty. The experimental results show that our proposed method have better performance compared with the state-of-the-art methods for fly species recognition.

The rest of this paper is organized as follows. Section II reviews the related work. Section III is the fly species recognition model. Section IV describes the experimental part of the fly species recognition. Section V outlines the experimental results and analyzes the feasibility of proposed method. Section VI introduces the conclusion and future research directions.

II. RELATED WORK

The recognition of fly species mainly depends on experts and technicians by observing the texture, color and morphological characteristics of flies. In the case of large numbers and types of flies, professionals need to spend a lot of manpower, material resources, and time. In the process of quarantine, it is difficult to accurately recognize the unknown fly species. Due to the development of international trade and the increasing number of inbound goods, the number of technical personnel required for customs quarantine cannot meet the actual needs and cannot provide technical support for rapid onsite law enforcement. With the development of information technology, the recognition of fly species adopts a computer-based automatic recognition method, which combines feature extraction and classifiers. However, this method has the problem of dependence on manual design features, resulting in inaccurate recognition results. Therefore, we use the method of deep learning to recognize fly species.

With the development of CNN, there are some feature extraction networks for object detection methods, such as VGG [26], ResNet and ResNeXt [27]. VGG was proposed by Simonyan et al. It increased the representation depth of the network by stacking some small convolution kernels and max-pooling layers. The advantage was that the structure was very simple, and the network performance could be improved by deepening the network structure. However, VGG uses more parameters, resulting in taking up more memories. ResNet adopted residual module and introduced an identity map to solve the degradation problem in the depth grid. It learnt the residual function $F(x) = H(x) - x$ by adding the residual path. The residual network used the shortcut connections method to directly pass the input x to the output as the initial result, which added neither extra parameter nor computational complexity. Compared with VGG, it has a deeper network and lower computational complexity. On the basis of ResNet, Xie et al. proposed ResNeXt, which used
the idea of increasing the cardinality. Compared with ResNet, it has the same parameters but higher accuracy. Therefore, we use ResNeXt as the feature extraction network in this paper.

Nowadays, attention models [28] have become an important field in the research of neural networks. The inspiration of attention mechanism can be attributed to people’s physiological perception of the environment. The attention model enables the network to extract information at key locations with less energy consumption, improving the performance of CNN. According to the relationship between feature channels, Hu et al. proposed SeNet [29], which was a typical deep learning method with attention mechanism. The Squeeze-and-Excitation module performed the squeeze operation on the convolved feature map to obtain channel-level global features. Then, it performed an excitation operation on the global features, which learnt the non-linear relationship between each channel and obtained the weights of different channels. Each channel weight coefficient was multiplied by the original feature map to obtain the final feature. This attention mechanism allows the model to focus on the channel features with the most information, while suppressing unimportant channel features. But it only pays attention to the relationship between the feature channels, does not make full use of global context information. Woo et al. proposed CBAM, which combined two dimensions of feature channel and feature space. On the basis of SeNet, the max-pooling feature extraction method was added, and the features of channel attention extraction were used as the input of the spatial attention module. This method not only saves parameters and computing power, but also brings stable performance improvements. We further improve on the basis of CBAM to improve the performance of the attention convolution module.

The loss function is used to estimate the degree of inconsistency between the predicted value and the true value of the model. The loss functions for object detection include classification loss functions and bounding box regression loss functions. The cross-entropy loss function was commonly used in classification tasks, which was good at learning information between classes. However, it only cares about the accuracy of predicting the probability of correct labels, and ignores the differences of other incorrect labels, resulting in poor learned features. Lin et al. proposed a focal loss based on cross entropy loss to solve the problem of class imbalance during training. Therefore, the classification loss function in this paper uses focal loss. In the regression loss function, smooth L1 loss function was generally used, which solved the problem of non-smoothness of L1 loss and the gradient explosion of L2 loss at the outliers. He et al. proposed a bounding box regression loss, namely KL loss, which combined with KL divergence to make the model fit better and improve the positioning accuracy. The KL loss takes into account the ambiguities of the ground-truth bounding box. Compared to smooth L1, its positioning is more accurate. Therefore, we try to adopt KL loss as the new bounding box regression loss function.

III. FLY SPECIES RECOGNITION MODEL
A. IMPROVEMENT OF RetinaNet MODEL
We use the RetinaNet model to recognize flies. RetinaNet model uses ResNeXt [16] as the feature extraction network, as shown in Figure 1 (a). The improved FPN is shown in Figure 1 (b). Bottom-up path augmentation is shown in Figure 1 (c). FCN [30] are used as classification subnet and regression subnet. These subnets are used for object classification and location, as shown in Figure 1 (d). Figure 1 shows the overall structure of the improved RetinaNet model.

RetinaNet uses ResNeXt as the feature extraction network, and ResNeXt is improved on the basis of the residual network ResNet. ResNeXt adopts a layer stacking strategy similar to VGG and ResNet networks, while adopting the idea of split-transform-merge [31,32] in a simple and scalable way. The input is distributed to some branches, and then these branches are weighted and summed. It does not increase the complexity of the parameters. ResNeXt reduces the number of hyper-parameters and uses the idea of increasing the cardinality, as shown in Equation (1). Each branch of ResNeXt uses the same topology, so the hyperparameters are reduced. For the conv3, conv4 and conv5 output of ResNeXt, the output of these last residual blocks is expressed as \( \{C3, C4, C5\} \).

\[
y = x + \sum_{i=1}^{Ca} T_i(x)
\]

where \( y \) is the output, \( x \) is input vector to the neuron, \( T_i(x) \) has the same topology which projects \( x \) to the subspace and converts it, \( Ca \) is the size of the set of transformations.

It adopts the FPN as the backbone network for RetinaNet. FPN uses top-down paths and horizontal connections to enhance the convolutional network, which obtains feature maps with different resolutions, including the semantic information of the deepest feature map. Therefore, FPN is a multi-scale feature pyramid. RetinaNet’s feature pyramid has 5 layers, including P3 to P7. P3 to P5 are obtained from the corresponding ResNeXt residual stage using top-down and horizontal connections. P6 and P7 are generated differently from FPN. P6 is obtained on the basis of C5 by \( 3 \times 3 \) stride-2 conv. After applying the ReLU [33] function on the basis of P6, P7 is obtained by the same \( 3 \times 3 \) stride-2 conv.

Both low-level and high-level features have their own advantages and disadvantages. The detail features of low-level features are rich, but the semantic information is less. The semantic information of high-level features is rich, but the location is rough. In real life, the size of fly images is different, so we need to improve the semantic information of low-level features and the locating ability of high-level features.

The connection method of FPN pays more attention to the resolution of adjacent layers and less attention to the resolution of other layers. Therefore, as shown in the blue line in Figure 1 (b), on the basis of FPN, we design CFPLF structure and add it to FPN, which enables the low-level features to obtain rich semantic information. The generation
Y. Chen et al.: Research on Recognition of Fly Species Based on Improved RetinaNet and CBAM

**FIGURE 1.** Improved RetinaNet architecture. (a)ResNeXt as feature extraction network. (b) The improved FPN (c) Bottom-up path augmentation. (d) Classification subnet and regression subnet.

**FIGURE 2.** (a) Illustration of CFPLF structure; (b) Illustration of bottom-up path augmentation process.

Processes of P5, P6 and P7 are the same. Taking P7 as an example, the specific process is shown in Figure 2 (a). We upsample the resolution of P7 by a factor of 4. The upsampled feature map and the C5 convolved feature map by element-wise addition. The fused feature map reduces the aliasing effect of upsampling through a $3 \times 3$ convolutional layer, and finally obtains P5.

Inspired by Liu et al. [25], we introduce a bottom-up path augmentation method. The purpose is to retain more low-level features and further enhance the positioning capability of the entire feature level. We use this method at the P3 to P7 layers, where $\{N_3, N_4, N_5, N_6, N_7\}$ represents the newly generated feature map corresponding to $\{P_3, P_4, P_5, P_6, P_7\}$. N3 is P3 without any operation. The generation processes of N5, N6, N7 and N4 are the same. Taking N4 as an example, the specific process is shown in Figure 2 (b). The feature map N3 passes a $3 \times 3$ stride-2 conv to reduce the space size. Then each element of the feature map P4 and the downsampled map are added through the lateral connection. The fused feature map passes through $3 \times 3$ convolutional layers to obtain N4. The channel of feature map is 256, and all convolutional layers are followed by ReLU activation function.

Classification subnet and regression subnet are small FCN attached to each FPN level. They have similar structures, but didn’t share parameters. The classification subnet applies four $3 \times 3$ convolutional layers, each with 256 filters and each followed by ReLU activation function. Then, each is followed by a $3 \times 3$ convolutional layer with $9 \times K$ filters, where $K$ is the number of object categories. Finally, sigmoid is used for output the KA binary predictions per spatial location. The regression subnet is in parallel with the classification subnet, which regresses the offset from each anchor to a nearby ground-truth box. We use the bounding box parameterization and regression methods from [23]. The KL loss function is obtained by the standard deviation of the predicted coordinates and the ground-truth bounding box. It reversely modifies the coordinate position of the box and the size of the predicted box.

### B. ATTENTION MODULE

CBAM pays attention to the important features and suppresses the unimportant features in the network, which effectively improves the performance of the CNN. To improve the performance of the RetinaNet model, an improved CBAM is added to the feature extraction network ResNeXt. CBAM includes channel attention module and spatial attention module. The channel attention module applies average-pooling and max-pooling to compress the spatial dimensions of feature maps, respectively. The spatial attention module applies average-pooling and max-pooling along the channel dimensions, respectively. Max-pooling considers only the largest element, ignores other elements in the pooling region, and retains more texture information of the image. Average-pooling calculates the average of all elements in the pooling region to retain more image background information. The process of max-pooling and average-pooling are shown in Figure 3. In this paper, we add stochastic-pooling [34] on this basis, and call the improved CBAM as Stochastic-CBAM. Stochastic-pooling assigns probabilities to the elements in the feature map according to the numerical value. The probability of element selection is positively related to the magnitude of the value. Stochastic-pooling can randomly
retain image information and reduce the loss of useful information. By using three pooling methods to extract features, we obtain a more complete attention map and better grasp the global information of the image receptive field.

Stochastic-pooling uses a stochastic process instead of a deterministic pool operation. It randomly selects elements according to the polynomial distribution given in the pool area. Max-pooling discards other useful information, and average-pooling will cause positive and negative activation values to cancel each other out. As shown in Equation (2), the probability of each area is calculated by normalizing the activation in the area.

$$z_i = \frac{a_i}{\sum_{k \in R_j} a_k}$$  \hspace{1cm} (2)$$

where $z_i$ is the probability of each area $j$, $R_j$ is the pooling region, $a_i$ is the element within $R_j$.

The locations within the area are sampled from a probability-based multinomial distribution. Stochastic-pooling is defined as Equation (3).

$$Y_j = A_f \text{ where } l \sim z(z_1, \ldots, z_{|R_j|})$$ \hspace{1cm} (3)

where $Y_j$ is the output of the pooling operator associate with the jth feature map, $A_f$ is the pooled activation.

The specific process of stochastic-pooling is shown in Figure 4. The elements in the pooling region are normalized to obtain a probability matrix, and a region is randomly selected according to the probability. The pooling value is the value of the selected region position. Therefore, stochastic-pooling can randomly retain more useful image information.

Stochastic-CBAM is a lightweight universal module, which can be integrated into a convolutional neural network for end-to-end training. It focuses on the important features and suppressed unnecessary features of the feature map, so it effectively improves the information flow in the network. We process the spatial and channel dimensions of the feature map through three pooling methods: average-pooling, max-pooling, and stochastic-pooling. Figure 5 shows the exact location of the Stochastic-CBAM inside the ResNeXt module. The structure in the black dotted frame is a block of ResNeXt with $C = 32$, where $C$ is the width of the input or output. After the intermediate feature map $F \in R^{C \times H \times W}$ as input, one-dimensional channel attention map $M_C \in R^{C \times 1 \times 1}$ is obtained through Stochastic-CBAM. $M_C$ is shown in Figure 6. We use the inter-channel relationships relationship of features to generate a channel attention map. Each channel of the feature map is considered as a feature detector. To efficiently calculate the channel attention, the spatial dimension of the input feature map is compressed. The channel attention module is shown in Figure 6. After inputting the feature map $F \in R^{C \times H \times W}$, we perform average-pooling, max-pooling, and stochastic-pooling on the feature map $F \in R^{C \times H \times W}$. We use different information to compress each channel of the feature map, and then get three channel attention vectors. Three channel attention vectors enter a shared network consisting of a hidden layer and multilayer perceptrons (MLP), which generate three attention vectors with dimensions $C \times 1 \times 1$. In order to reduce the cost of calculation parameters, the hidden activation size is set to $R_{C/r \times 1 \times 1}$ where $r$ is the reduction rate and the activation function is ReLU. The output layer is increased to $C$ to obtain the same number of feature vectors as the number of feature map channels. Finally, the corresponding positions of the three feature vectors are summed. Channel attention map $M_C(F)$ is shown in Equation (4).

$$M_c(F) = \Delta \left( W_1 (W_0 (F_{avg})) \right) + W_1 \left( (W_0 (F_{max})) + W_1 \left( (W_0 (F_{sto})) \right) \right)$$  \hspace{1cm} (4)

where $W_0$ and $W_1$ are the weight of MLP, and $\Delta$ is the sigmoid activation function. $F_{avg}$ is the average-pooling of feature $F$. $F_{max}$ is the max-pooling of feature $F$, and $F_{sto}$ is the stochastic-pooling of feature $F$.

Spatial attention map is generated by using the spatial relationship of features. Spatial attention is different from channel attention, which focuses on distinguishing the position of features and complementing channel attention. The Spatial attention module is shown in Figure 7. After obtaining the optimized feature map $F'$ with the channel attention map, we also use average-pooling, max-pooling, and stochastic-pooling for each channel. Then, we make use of three pooling operations to generate three feature maps with the same dimensionality, which are stitched together to obtain special feature map. Finally, we perform a convolution operation with the filter size of $7 \times 7$ on this special feature map, and then generate a two-dimensional spatial attention map $M_S(F')$ through the sigmoid function. The expression
FIGURE 5. Internal structure of ResNeXt module based on Stochastic-CBAM.

FIGURE 6. Illustration of channel attention module.

FIGURE 7. Illustration of spatial attention module.

The loss function of the RetinaNet network consists of two parts: the classification loss function and the bounding box regression loss function. We use the focus loss as the classification loss to solve the problem of class imbalance, and use the KL loss as the bounding box regression loss to consider the ambiguities of the ground-truth bounding box. The loss function is shown in Equation (6).

$$L = \frac{1}{N_{cls}} \sum_i L_{cls} + \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}$$  \hspace{1cm} (6)$$

where $L$ is the loss function, $L_{cls}$ is the classification loss function, and $L_{reg}$ is the bounding box loss function. $N_{cls}$ represents the number of classification samples, $N_{reg}$ represents the number of regression samples, $\lambda$ is the weight coefficient, $i$ is the anchor point, $p_i^*$ is the anchor discrimination value. When the anchor point is a positive sample, $p_i^*$ is 1. When the anchor point is a positive sample, $p_i^*$ is 0.

Due to the problem of class imbalance in the training process, we use the focal loss function. In order to focus on hard samples and reduce the attention on easy samples, the focal
loss function adds a modulation factor of $-\alpha_i (1 - p_i)^\gamma$ to cross entropy loss function. The focal loss function is given by Equation (7). $p_i$ is shown in Equation (8).

$$L_{cls} = -\alpha_i (1 - z_i)^\gamma \log (z_i)$$ (7)

$$p_i = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases}$$ (8)

where $p_i$ is the prediction probability of the binary classification; $\alpha_i$ is the weighting factor, which is used to adjust the proportion of positive and negative samples; $\gamma$ is the focusing parameter, which adjusts the weighting ratio of the easy samples. $p$ is the model prediction probability. $y$ is the ground-truth label.

To simulate the situation where the flies are blocked by other objects in the natural environment, we perform occlusion and partial blur processing on the fly images. In the regression task, smooth L1 loss does not consider the ambiguity of the ground-truth bounding box. Therefore, we use KL loss function instead of smooth L1 loss function to learn the bounding box regression and positioning uncertainty. We can represent the ground-truth bounding box as a Gaussian distribution, and independently optimize the coordinates $x$ of each bounding box. The regression loss is regarded as the KL divergence between the Gaussian distribution and the Dirac delta function. $L_{reg}$ is defined as shown in Equation (9).

$$L_{reg} = D_{KL} (P_D (x) || P_\theta (x))$$ (9)

where $D_{KL}$ is KL divergence, $P_D$ is Dirac delta function, and $P_\theta$ is single variable Gaussian function.

The bounding box regression loss function of a single sample is defined as shown in Equation (10). To avoid gradient exploding, our network predicts $\beta = \log (\sigma^2)$ instead of $\sigma$.

$$L_{reg} \propto e^{-\beta (\left| x_g - x_e \right| - \frac{1}{2})^2 + \frac{1}{2} \beta}$$ (10)

where $x_g$ is the ground-truth bounding box location, $x_e$ is the estimated bounding box location, and $\sigma$ is the standard deviation.

For $\left| x_g - x_e \right| > 1$, the bounding box regression loss function of a single sample is defined as shown in Equation (11).

$$L_{reg} = e^{-\beta \left( \left| x_g - x_e \right| - \frac{1}{2} \right)^2 + \frac{1}{2} \beta}$$ (11)

IV. EXPERIMENT

The process of fly species recognition is shown in the Figure 8. Original images of flies were obtained by an image acquisition device. A large number of images were obtained by data augmentation processing, and then the fly dataset was made. We trained the recognition model on the fly dataset, and then flies were recognized through the trained model.
COOLPIX A1000 digital camera. The image resolution was 4608 × 3456 pixels with the jpg image format. Figure 9 shows the image acquisition device. Figure 10 shows some original image samples.

A. DATASET
The flies in this paper were from the China Customs Laboratory. The flies had been made into specimens to ensure that their shape and color didn’t change. There were eleven types of flies, such as Lucilia caesar (Lc), Chrysomya megacephala (Cm), Neomyia laevifrons (Nl), Anthomyia illocata (Ai), Musca domestica L (MdL), Muscina stabulans (Ms), Aldrichina grahami (Ag), Lucilia illustris (Li); Lucilia sericata (Lse), and Boettcherisca peregrina (Bp). The image shooting equipment was a Nikon COOLPIX A1000 digital camera. The image resolution was 4608 × 3456 pixels with the jpg image format. Figure 9 shows the image acquisition device. Figure 10 shows some original image samples in the laboratory.

In this experiment, it was necessary to make a sample dataset for the fly species recognition. There are 11 species of flies, and each species has 30 flies. Starting from the head of the fly, we rotated the fly clockwise and took an image every 90°. A total of 2640 images were obtained. Due to the high resolution of the image, the training time of the dataset was long. We resized the size of the image to 500 × 375. The increase in the amount of training data improved the generalization ability of the model and avoided overfitting due to little data. We enhanced the dataset while ensuring the same number of each species. All images were rotated every 90°, and some images were scaled, partially blurred and occluded. Then, we obtained a dataset containing 15840 images. In the experiment, we randomly selected 11088 images as the training set, 2376 images as the validation set, and 2376 images as the test set. Then, we applied the labelImg-image-labeling tool to label the outlines of flies in the dataset.

B. MODEL TRAINING
The experiment computer was configured with an Intel Core i7-8700 processor, 64GB memory and 1TB hard disk, and the graphics card was NVIDIA GeForce RTX 2080TI. The training weight of ResNeXt on the ImageNet dataset [35] was used as the pre-training parameter of the model. Then, an improved attention module was added to ResNeXt. During training, we used stochastic gradient descent algorithm to optimize the training model. After comparative experiments, we used an initial learning rate of 0.001, a weight decay of 0.0001 and a momentum factor of 0.9. We set the initial number of training epoch to 81. Epoch means that all samples in the training set were trained once. When the number of iterations reached the 45th epoch, the learning rate dropped to 10% of the previous stage. When the number of iterations reached the 65th epoch, the learning rate dropped to 10% of the previous stage. The training image batch size was 32. In comparative experiments, we found that the weighting factor $\alpha_i$ of 0.25 and the focusing parameter $\gamma$ of 2.5 in the focus loss are the best. In KL loss function, the standard deviation and mean were set to 0.0001 and 0 respectively. When the IoU of the anchor and ground-truth object box was greater than 0.5, the anchor was a positive sample; when the IoU was less than 0.4, the anchor was a negative sample. The NMS threshold was 0.5, which aimed to remove redundant bounding boxes.

We used different bounding box regression loss functions to generate the training loss curve as shown in Figure 11. The two different bounding box regression loss functions were smooth L1 loss and KL loss, and the classification loss function was focal loss. Training loss was the sum of classification loss and bounding box regression loss. At the beginning of training, the training loss generated by the two methods drops rapidly. During the training process, the KL loss function as the bounding box regression loss produced less shock. The results showed that the training loss of the two methods converged and tended to a stable value with the training epoch, and the training process is ideal. We tested the trained model on the test set to verify the effect of different bounding box loss functions on fly species recognition.

C. EXPERIMENTAL EVALUATION INDEX
We evaluated the performance of the model by analyzing the Average Precision (AP) and mAP of the experimental results. We plotted the precision-recall curve with the precision as the ordinate and the recall as the abscissa. The AP value was the area between the curve and the coordinate axis. The mAP value was the average AP value of multiple categories. mAP was used to measure the performance of the model on all categories. The closer the AP and mAP values were to 1, the better the target detection performance of the model was. Precision referred to the proportion of positive samples predicted correctly to all predicted positive samples. Recall...
**D. RESULTS AND ANALYSIS**

To verify the performance of the improved RetinaNet network for fly species recognition, we tested 2376 images in the trained model. It includes 11 species of flies, and each species of flies has 216 images. Figure 12-14 shows the recognition results of the head, tail and side of some flies, respectively.

Figure (a), (b), (c), and (d) are the correct recognition images, and figure (e) is the wrong recognition images. As shown in Figure 12 (e), Boettcherisca peregrina is mistakenly identified as Neomyia laevifrons during the head recognition. The main reason is that they have similar head shapes and brown eyes. Figure 13 (e) shows that Lucilia shenyangensis is mistakenly identified as Lucilia caesar during the tail recognition. The tails are similar in color and shape, so they are difficult to identify. Also in Figure 14 (e), Lucilia illustris is mistakenly identified as Chrysomya megacephala during side recognition. They are difficult to identify due to the similar side shape and color.

1) **COMPARISON OF FEATURE EXTRACTION NETWORKS**

Table 1 shows the performance of fly species recognition in different feature extraction networks. The proposed model is compared with the RetinaNet network model based on three other feature extraction networks. We obtain the AP, mAP, and detection speed of flies, as shown in Table 1. Through the analysis of the experimental results, the mAP of the proposed model is 90.38%. The mAP value of the proposed model is 2.61% higher than the RetinaNet model based on ResNeXt101 with CBAM (87.77%). The mAP value of the proposed model is 6.02% and 10.48% higher than the RetinaNet model based on ResNeXt101 (84.36%) and ResNet101 (79.90%), respectively, which means it has the best performance. In this paper, the time for the model to recognize each image is 0.131s, so the detection speed is not significantly reduced. Experimental results show that ResNeXt101 improves the average accuracy of fly species recognition compared to ResNet101. ResNeXt101 adopts the idea of increasing the cardinality without increasing the complexity of the network. Its branches use the same topology. We add the improved CBAM to the feature extraction network. It takes advantage of the channel and spatial relationships of features to further improve the performance of the model. Compared the ResNeXt101 with the ResNeXt101 with CBAM, we get that the difference of mAP value between the two is 3.41%, and the recognition time of a single image differs by only 0.024s. Figure 15 shows the visualization of feature maps of three feature extraction networks. We can see from the figure that CBAM-integrated network (ResNeXt101 + CBAM) retains more pixel information and texture features than ResNeXt. Our method and CBAM-integrated network retain similar pixel information of feature maps, but our method produces finer attention.
TABLE 1. Comparison of feature extraction networks.

| Feature extraction network | Lc  | Cm  | Nl  | Ai  | MdL | Ms  | Li  | Ls  | Lse | Bp  | mAP  | Test time of a single image/s |
|----------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|-----------------------------|
| ResNeXt101                 | 85.27 | 86.24 | 86.67 | 88.14 | 87.92 | 83.35 | 82.08 | 78.20 | 80.02 | 83.70 | 86.41 | 84.36 | 0.103                     |
| ResNet101                  | 79.90 | 80.51 | 80.79 | 83.83 | 85.32 | 76.07 | 77.89 | 80.18 | 75.43 | 78.44 | 80.49 | 79.90 | 0.095                     |
| ResNeXt101+CBAM           | 90.02 | 87.25 | 88.78 | 88.39 | 90.01 | 89.27 | 86.56 | 83.52 | 84.50 | 87.39 | 89.73 | 87.77 | 0.127                     |
| Proposed                   | 90.76 | 92.43 | 93.13 | 84.35 | 94.23 | 93.42 | 89.54 | 82.27 | 89.23 | 92.55 | 92.24 | 90.38 | 0.131                     |

FIGURE 15. Feature map visualization of feature extraction network.

TABLE 2. Comparison of the performance of different components on fly species recognition. BPA represents bottom-up path augmentation.

| Smooth L1 | KL | FPN | Improved FPN | BPA | mAP/% |
|-----------|----|-----|--------------|-----|-------|
| ✓          | ✓  | ✓   | ✓            | ✓   | 78.29 |
| ✓          | ✓  | ✓   | ✓            | ✓   | 77.04 |
| ✓          | ✓  | ✓   | ✓            | ✓   | 82.27 |
| ✓          | ✓  | ✓   | ✓            | ✓   | 84.51 |
| ✓          | ✓  | ✓   | ✓            | ✓   | 84.06 |
| ✓          | ✓  | ✓   | ✓            | ✓   | 90.38 |

3) COMPARISON OF INITIAL LEARNING RATE

In the stochastic gradient descent method, the initial learning rate plays an important role in the convergence of the deep network. We choose an update strategy with a good learning rate, which can reach the minimum value of loss faster, and the converged loss value is the global optimal solution of the neural network. In order to better study the effect of learning rate on the fly species recognition, we conducted a comparative experiment. We set different initial learning rates, and the learning rate decreased by 10 times in the same training epoch. The best initial learning rate is determined by mAP of recognition, as shown in Table 3, where Lr represents the initial learning rate. When the initial learning rate is 0.001, the mAP is the best. When the initial learning rate is greater than or less than 0.001, the result of recognition will be affected. A large learning rate easily leads to a large fluctuation of the cost function, so it is difficult to find the global optimal point. A small learning rate leads to a slow change of the loss function and increases the convergence complexity of the network.

4) HYPERPARAMETRIC ANALYSIS OF FOCAL LOSS

Focal loss solves the problem of class imbalance. We set up comparative experiments to obtain better weight main reason is that the CFPLF structure is added on the basis of FPN, which enriches the shallow semantic information. When using the KL loss function, by adding the bottom-up path augmentation method, the mAP of fly species recognition increased by 6.32%. This verifies the usefulness of information from lower feature levels. In summary, the method proposed in this paper greatly improves the effect of fly species recognition.

2) COMPONENT ABLATION STUDIES

In the experiment, we analyzed the effect of some component on fly species recognition, including improved FPN, and bottom-up path augmentation and KL loss. The experimental results are shown in Table 2. We use the improved feature extraction network in the experiment. Since we simulated the situation where the flies are blocked by other objects in the natural environment, compared with the smooth L1 loss, the mAP obtained by using the KL loss function is improved by 8.11%. Compared with the traditional FPN, the improved FPN improves the mAP for the recognition of fly species. The
parameters $\alpha_i$ and focusing parameters $\gamma$ in this paper. When other parameters are the same, the $\alpha_i$ and $\gamma$ are different. According to the obtained mAP, the better hyperparameters for model training are selected. The comparison results are shown in Table 4. When $\alpha_i$ is 0.25 and $\gamma$ is 2.5, the mAP is the best. Compared to the original RetinaNet, the weighting factor is the same, and the focusing parameter is increased by 0.5. Compared with the weight factor of 0.25 and the focusing parameter of 2.0, the mAP of this paper increased by 2.78%.

5) ANALYSIS OF TRANSFER LEARNING

This paper uses ResNeXt pre-trained on ImageNet as the feature extraction network. To understand the importance of transfer learning for this paper, we conduct a comparative experiment. When other parameters are the same, we compare the training loss function and mAP. As shown in Table 5, in the method of transfer learning, the mAP increased by 14.86%, and the final training loss function decreased by 0.21. Through experimental analysis, the method of transfer learning improves the performance of the model and has higher recognition accuracy.

6) COMPARISON WITH OTHER STATE-OF-THE-ART METHODS

We compare the proposed method with other state-of-the-art methods for fly species recognition, in order to better analyze the performance of the network. The results are shown in Table 6. We find that the mAP of this paper is 4.26% higher than Mask R-CNN (86.12%), 19.82% higher than YOLOv3 (70.56%) and 2.29% higher than M2Det (88.09%). Compared with the original RetinaNet (70.64%), our proposed method has a better effect on fly species recognition. Compared with the method of recognizing different objects with similar characteristics, the mAP of our proposed method is 2.01% higher than TASN (88.37%) and 0.74% higher than WS-DAN (89.64%). Therefore, compared with other state-of-the-art methods, the results of our method are better.

Table 2 also shows that the average time for the method proposed in this paper to recognize a single image is 0.131s. The proposed method is 0.078s faster than Mask R-CNN. It is 0.077s and 0.058s slower than YOLOv3 and M2Det, respectively. Our method takes less test time than TASN and WS-DAN. The experimental results show that our proposed method has higher accuracy in recognizing fly species compared with other state-of-the-art methods.

V. CONCLUSION

In this paper, we propose a fly recognition method based on improved RetinaNet and CBAM, which accurately locates and recognizes flies. We designed the CFPLF structure and introduced a bottom-up path augmentation to improve the low-level features semantic information and the high-level features location ability. We improved CBAM in order to effectively improve the performance of CNN, and used KL loss to deal with the ambiguities of the ground-truth bounding box. The mAP of our proposed method to recognize fly species is 90.38%. The experimental results show that our proposed method have better performance compared with the state-of-the-art methods for fly species recognition. This is of great significance for the fly species recognition. In future work, we will further study the methods of fly species recognition. We will continue to improve the accuracy of fly species recognition and improve the detection speed.

REFERENCES

[1] Q. Yao, J. Lv, Q.-J. Liu, G.-J. Diao, B.-J. Yang, H.-M. Chen, and J. Tang, “An insect imaging system to automate rice light-trap pest identification,” J. Integrative Agric., vol. 11, no. 6, pp. 978–985, Jun. 2012, doi: 10.1016/S2095-3119(12)60089-6.

[2] S.-H. Kang, J.-H. Cho, and S.-H. Lee, “Identification of butterfly based on their shapes when viewed from different angles using an artificial neural network,” J. Asia Pacific Entomol., vol. 17, no. 2, pp. 143–149, Jun. 2014, doi: 10.1016/j.aspen.2013.12.004.
[3] C. Wen, D. E. Guyer, and W. Li, “Local feature-based identification and classification for orchard insects,” Biosyst. Eng., vol. 104, no. 3, pp. 299–307, Nov. 2009, doi: 10.1016/jbiosystemseng.2009.07.002.

[4] J. Wang, C. Lin, L. Ji, and A. Liang, “A new automatic identification system of insect images at the order level,” Knowl.-Based Syst., vol. 33, pp. 102–110, Sep. 2012, doi: 10.1016/j.knosys.2012.03.014.

[5] J. Tang, F. Gao, F. Liu, and X. Chen, “A denoising scheme-based traffic flow prediction model: Combination of ensemble empirical mode decomposition and fuzzy C-Means neural network,” IEEE Access, vol. 8, pp. 11546–11559, 2020.

[6] X. Chen, S. Wang, C. Shi, H. Wu, J. Zhao, and J. Fu, “Robust ship tracking via multi-view learning and sparse representation,” J. Navigat., vol. 72, no. 1, pp. 176–192, Jan. 2019, doi: 10.1017/S0373463318000952.

[7] X. Chen, Y. Yang, S. Wang, H. Wu, J. Tang, J. Zhao, and Z. Wang, “Ship type recognition via a coarse-to-fine cascaded convolution neural network,” J. Navigat., pp. 1–20, Feb. 2020, doi: 10.1017/S0373463319000990.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Lake Tahoe, NV, USA, Dec. 2012, pp. 1097–1105.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Columbus, OH, USA, Jun. 2014, pp. 580–587.

[10] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Boston, MA, USA, Dec. 2015, pp. 1440–1448.

[11] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy, Oct. 2017, pp. 2961–2969.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[13] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 2117–2125.

[14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., Montreal, QC, Canada, Dec. 2015, pp. 91–99.

[15] Z. Cai and N. Vasconcelos, “Cascade R-CNN: Delving into high quality object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 6154–6162.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–788.

[17] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Honolulu, HI, USA, Jul. 2017, pp. 7263–7271.

[18] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, arXiv:1804.02767. [Online]. Available: http://arxiv.org/abs/1804.02767.

[19] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, “M2Det: A single-shot object detector based on multi-level feature pyramid network,” in Proc. AAAI Conf. Artif. Intell., Honolulu, HI, USA, Jan. 2019, pp. 9259–9266.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy, Oct. 2017, pp. 2980–2988.

[21] H. Zheng, J. Fu, Z.-J. Zha, and J. Luo, “Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Long Beach, CA, USA, Jun. 2019, pp. 5012–5021.

[22] T. Hu, H. Qi, Q. Huang, and Y. Lu, “See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification,” 2019, arXiv:1901.09891. [Online]. Available: http://arxiv.org/abs/1901.09891.

[23] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, Sep. 2018, pp. 3–19.

[24] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, “Bounding box regression with uncertainty for accurate object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Long Beach, CA, USA, Jun. 2019, pp. 2888–2897.

[25] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 8759–8768.

[26] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556.

[27] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 1492–1500.

[28] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 3156–3164.

[29] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.

[30] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Boston, MA, USA, Jun. 2015, pp. 3431–3440.

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Boston, Jun. 2015, pp. 1–9.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 2818–2826.

[33] G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in Proc. Int. Conf. Mach. Learn., Haifa, Israel, Jun. 2010, pp. 807–814.

[34] M. D. Zeiler and R. Fergus, “Stochastic pooling for regularization of deep convolutional neural networks,” 2013, arXiv:1301.3557. [Online]. Available: http://arxiv.org/abs/1301.3557.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Miami, FL, USA, Jun. 2009, pp. 248–255.

YANTONG CHEN was born in Shenyang, Liaoning, China, in 1989. He received the B.Eng. degree in communication engineering from Jilin University, and the M.S. and Ph.D. degrees in mechanical and electronic engineering from the University of Chinese Academy of Sciences, in 2017. He is the author of ten articles. His research interests include remote sensing image processing, target recognition, and deep learning. He is a member of the Chinese Society of Naval Architects and Marine Engineers.

XIANZHONG ZHANG was born in jining, Shandong, China, in 1995. He received the B.Eng. degree in electronic information science and technology from the Shandong University of Technology, China, in 2019. He is currently pursuing the M.S. degree with Dalian Maritime University, China. His research interests include image processing and deep learning.
WEINAN CHEN was born in Hunchun, Jilin, China, in 1996. She received the B.Eng. degree in communication engineering from the Changchun University of Technology, China, in 2019. She is currently pursuing the M.S. degree with Dalian Maritime University, China.

Her research interests include target recognition and deep learning.

YUYANG LI was born in Liaoyang, Liaoning, China, in 1995. She received the B.Eng. degree in communication engineering from Dalian Ocean University, China, in 2017. She is currently pursuing the M.S. degree with Dalian Maritime University, China.

Her research interests include remote sensing image processing, target recognition, and deep learning.

JUNSHENG WANG received the B.Eng., M.S., and Ph.D. degrees in mechanical and electronic engineering from the Harbin Institute of Technology.

From 2011 to 2012, he was a Visiting Scholar with the Department of Mechanical and Electrical Engineering, University of Waterloo, Canada. His research interests include embedded system design, signal processing technology, and target recognition. He is the author of 55 articles, and hosted 17 research projects. He was selected as a transportation youth technology talent, one of the 100 Million Talent Projects, Liaoning, and so on.