Forest Chemicals Review
www.forestchemicalsreview.com
ISSN: 1520-0191
July-August 2021 Page No.135-162
Article History: Received: 10 May 2021 Revised: 20 June 2021 Accepted: 18 July 2021 Publication: 31 August 2021

FED: A Method for Detecting Indoor People Looking down under a Fisheye Lens

Rui Li*, Boxuan Yan
School of Computer and Communication, Lanzhou University of Technology, Lanzhou, Gansu, China
*Corresponding Author.

Abstract:

The Faster Region-based Convolutional Neural Network (Faster R-CNN) object detection model has become a milestone algorithm in deep learning due to its high precision and low computational complexity, but it still has certain limitations. In order to be suitable for indoor people detection under the fisheye lens, this paper proposes a fast object detection method, Fisheyes Effective Detection (FED) based on Faster R-CNN. First, Ameliorate Intersection over Union (AIoU) is used to improve the accuracy of the positioning phase. Secondly, using the ResNet with Residual-I module and adding the channel attention mechanism to extract more useful object feature information and reduce detection errors. Finally, when constructing the network, the FED detection network is constructed by changing the stacking order and replacing the activation function. Experimental results show that FED has superior detection performance, faster detection speed, and object recognition performance. It is more suitable for indoor personnel detection under a fisheye lens.

Keywords: *Faster R-CNN, AIoU, Attention, Overlooking the fisheye lens, Personnel detection.*

I. INTRODUCTION

With the development of deep learning, computer vision has been successfully applied in real life, such as intelligent security, pilotless driving, and other fields. As the core research problem of computer vision, object detection has attracted more and more attention from researchers [1]. Visual environment perception plays an indispensable role in the development of intelligent security. It can make the security system not rely on human visual judgment, and the system can automatically detect the monitoring screen. In order to achieve large-space monitoring, the typical method is to install multiple wide-angle, standard lenses above the scene. The other is to use a single overhead fisheye camera or a camera with a 360° field of view in the scene. The fisheye image pedestrian detection is different from standard pedestrian detection. Due to its unique radial distortion and barrel distortion, the human detection algorithm developed for the standard lens has a poor detection effect on the fisheye image. For people detection tasks under fisheye images, the conventional method is to perform distortion correction on the image before processing. That is, the circular perspective fisheye image is corrected into a two-dimensional planar vision image and then processed by the standard object detection algorithm [2]. The existence of the correction step leads to reduced detection real-time performance.

People photographed under a standard camera usually exist in an upright posture [3]. The conventional detection algorithm works well, but it does not perform well under a fisheye lens because the posture of the human body will be disturbed by distortion and present a non-upright state [4], which reduces the detection accuracy as shown in Fig 1.

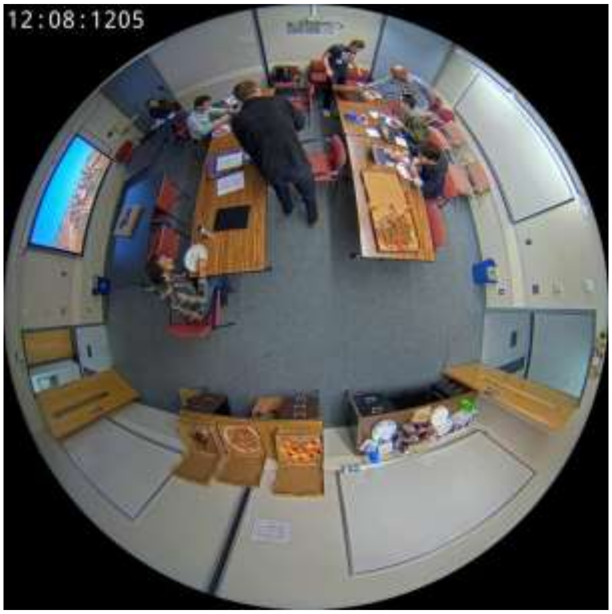


Fig 1. The characters in the picture are non-upright

In response to this problem, this paper is based on the challenging events of person detection in the elevated fisheye image (CEPDOF) dataset [5], and uses the improved neural network model based on Faster R-CNN [6] to detect fisheye from above people under the camera. Compared with the improvement of existing algorithms in this field, the model proposed in this paper does not require pre-processing or post-processing of pictures, and the time complexity is significantly reduced; a new feature extraction module is designed, channel attention is added, and the recognition accuracy is significant Increase; build an end-to-end model so that you can train or fine-tune the weights on the labeled fisheye images.

The main contributions of this paper are summarized as follows:

● A two-stage and end-to-end neural network model, FED, is proposed to detect people in fisheye images, proving that the model is simple and effective, and superior to existing methods.

● Compared with Faster R-CNN and RAPID [5] on the international general target detection data set CEPDOF, FED has excellent results, higher accuracy, and shorter inference time on a single GPU.

● The research on the detection of people under the fisheye lens can promote the development of intelligent security, which has high practical significance, and there is still room for use in follow-up research.

II. RELATED WORK

This section introduces the background and work associated with the proposed method.

2.1 Object Detection

In recent years, convolutional neural networks have achieved remarkable success in computer vision tasks such as image classification, semantic segmentation, and object detection [7]. Each layer of the convolutional neural network can extract features of the two-dimensional input information, and propagate the features to the subsequent network, and then perform the next step of processing the image. Compared with traditional detection algorithms, the convolutional neural network can be more efficient. It processes the two-dimensional local information of the picture, extracts the picture features, performs detection, and trains the model utilizing gradient descent and error backpropagation through a large amount of labeled data input.

At present, the resolution of images taken with a fisheye lens is usually 1024*1024 to 2048*2048. For large-resolution pictures, preprocessing is usually performed before input to reduce the size, but it will reduce the feature level of some of the images. The full convolutional network (FCN) proposed by Long J *et al*. [8] provides a new idea, using a fully convolutional network without a fully connected layer, which can be applied to any size input. Palla *et al*. [9] took the RGB-D image as input and generated the initial representation of the occlusion network from the depth channel. The full convolutional network was used to fill in the holes to generate a complete 3D model that is not subject to fixation and successfully reconstructed scenes of any size.

In deep learning, the inference is to apply the abilities learned in training to work. After carefully adjusting the weights, the neural network is a cumbersome and colossal database. In order to make full use of the training results and realize the detection task, it is necessary to retain the intellectual ability and quickly apply it to the data. Then, accelerating inference has become a research hotspot. Picture size and reasoning time are positively correlated. In the existing work of accelerating reasoning time, there are mainly two methods to optimize the neural network to achieve high speed and low latency. The first is to accelerate inference by improving hardware. Mas J, *et al*. [10] accelerate inference by building an NCS2 hardware cluster, using three NCS2 multi-processing acceleration hardware devices, and using the horizontal scalability of hardware to improve performance and reduce of reasoning time. The other is to speed up inference through network adjustment, fuse multiple-layer parameters in the neural network, or change the layer stacking method, and perform parameter fusion operations on the parts of the neural network that have not been reached or have not been activated after training to accelerate inference and improve network performance.

2.2 Traditional Camera Personnel Detection Algorithm

Among the traditional standard camera personnel detection algorithms, a representative one is the pedestrian detection algorithm proposed by Fe Navneet *et al.* [11] based on the combination of the histogram of oriented gradient (HOG) features and support vector machine (SVM) classifiers. HOG uses gradients to describe image information, traverses, and concatenates information that obtains the object characteristics. Edgar Seemann *et al*. [12] proposed a pedestrian detection algorithm based on the combination of DPM features and latent SVM classifiers for dense and occluded scenes. DPM is an extension of HOG, but many improvements have been made on the model, and the detection performance has been slightly improved. In recent years, algorithms based on deep learning have performed well in object and pedestrian detection tasks. The algorithm can be divided into two categories: two-stage and one-stage. Two-stage methods, such as region-based convolutional neural network (R-CNN) and its improved models, including a region of interest (ROI), region proposal network (RPN), and optimized Bounding Box (Bbox) network head. Among them, Faster R-CNN implements two-stage end-to-end training, which is a breakthrough in the two stages. For example, Single Shot Multi-Box Detector (SSD) [13] and You Only Look Once (YOLO) [14] are all one-stage methods, which are detected by abstracting the network as an independent RPN. A one-stage method is used for an input image to directly regress the bounding box through a convolutional neural network (CNN). At present, the main research goal is to build a fast and accurate object detector.

2.3 Indoor Personnel Detection

Article History: Received: 10 May 2021 Revised: 20 June 2021 Accepted: 18 July 2021 Publication: 31 August 2021

Most of the cameras for indoor personnel detection are from a bird's-eye view. The personnel background is quite different from that of the outdoor, and the posture of the human body is different, resulting in unsatisfactory conventional detection results. Wang Xia *et al*. [15] proposed a multi-model joint learning indoor personnel detection algorithm by integrating RPN into the Vgg-16 framework [16]. However, it does not support end-to-end training, and the accuracy is not enough to realize real-world applications. Kaiming *et al*. [17] proposed the Focal loss function to solve the imbalance of positive and negative samples in the network to a certain extent, making ResNet one of the cutting-edge models in the object detection network. However, it is also not suitable for indoor personnel detection.

2.4 Fisheye Image to Detect Indoor People

Looking down on indoor people under the fisheye lens is an emerging detection task. The existing algorithms only slightly modify the HOG and DPM models to match the geometric distortion of the human body and can be applied to fisheye images to complete the human detection task. Chiang and Wang *et al*. [18] performed a slight angle rotation on the fisheye image for detection, extracted the HOG feature from the center of the image, and used the SVM classifier for detection. Pedro *et al*. [19] used a component-based detection algorithm to model the combination of different parts of the person in the fisheye image and used the latent SVM classifier to complete the detection task. The Aggregate Channel Features (ACF) algorithm [20] is a detection algorithm based on multi-channel features. Since the features are directly extracted from the pixels in the down-sampling channel, the amount of calculation is less in comparison, but small object is ignored. This leads to a decrease in detection accuracy.

Recently, CNN has also been applied to this problem. Tamura *et al*. [21] designed a rotation-invariant version of YOLO by training the network on the rotating version of the COCO dataset [22]. This method relies on the assumption that the image Bbox and the radius are aligned in reasoning and uses the center point of the object and the image to determine the position of the distorted image. Duan *et al*. [5] improved the loss function of YOLO to detect the rotation of the human body, and judge the regression angle by the distortion angle of the human body in the picture, to achieve the purpose of human detection, but it is not universal. There is no angle value in the object's label in general data sets, such as COCO and VOC. Although the algorithm is accurate, it is computationally complex and can only object a specific data set, and cannot be widely used in fisheye lens detection. Xu Y *et al*. [23] used sliding vertices on the horizontal bounding box and achieved excellent results in multi-directional object detection. However, the human body is in a distorted state in the fisheye image, and the vertex sliding cannot perfectly surround the object, so the detection accuracy is slightly worse. Xu Shu *et al*. [24] developed a novel adaptive weighting model that optimizes the human body in the deformed image through the advantages of a part-based convolutional network in feature representation. It has a high accuracy rate, but it is suitable for standard cameras.

III. POSITIONING STAGE

3.1 Ameliorate Intersection over Union

When dealing with regression tasks, intersection over union (IoU) is the most commonly used indicator in object detection. It can determine positive and negative samples and determine the distance between predicting Bbox and ground-truth (GT), such as the formula (1) shown.

$$IoU = \frac{area(ROI_T \cap ROI_G)}{area(ROI_T \cup ROI_G)} \tag{1}$$

Formula (1), $area$ represents the area size, ROI_T is the prediction frame, and ROI_G is the real frame.

In the detection task, the precise positioning depends on the regression accuracy of the Bbox. However, the IoU loss based on the 1-norm or the 2-norm leads to a slightly worse prediction effect. The Generalized Intersection over Union (GIoU) proposed by Tsoi N, *et al*. [25] tends to let Anchor and the target frame produce a larger intersection area. Then GIoU degenerates into IoU regression strategy. Therefore, the speed will be very slow and there is a certain probability cause the results to diverge. When GT Bbox contains predict Bbox, GIOU's convergence effect in the horizontal and vertical directions is extremely poor. The reason is that the gradient does not penalize the two directions enough, which leads to divergence of the results.

In order to solve the above problems, this article has improved the calculation method of GIoU and named it AIoU (Ameliorate IoU). Normalize the distance between the maximum and minimum coordinates of the anchor point and the target frame (minimize the coordinate distance), and add penalty items based on GIoU (provide vertical and horizontal gradient changes). When the prediction frame and the target frame do not overlap, the prediction frame will move to the target frame to obtain a better convergence effect. The schematic diagram is shown in Fig 2.

Article History: Received: 10 May 2021 Revised: 20 June 2021 Accepted: 18 July 2021 Publication: 31 August 2021

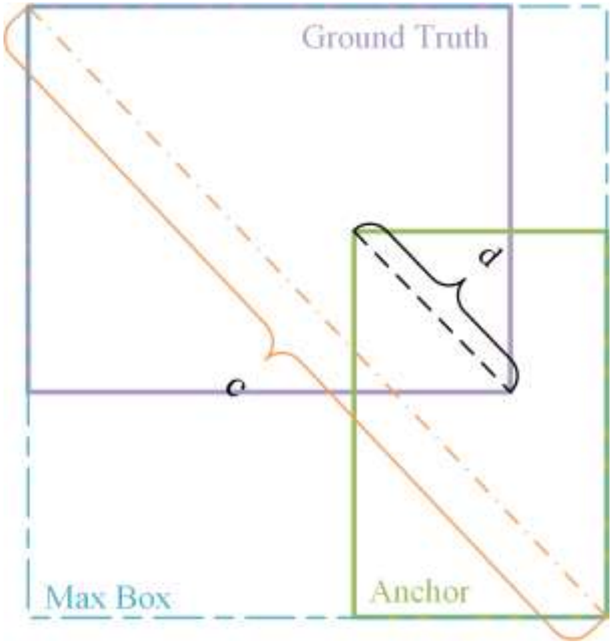


Fig 2. AIoU schematic

The definition and calculation of AIoU are very simple. When calculating the overlapping area of two rectangular boxes, you only need to calculate the maximum and minimum values of the coordinates of the two boxes (the upper left corner and the lower right corner coordinates), surrounded by these two coordinates The closed area (rectangular box) is defined as A^{in}, The distance between the maximum and minimum coordinates is defined as d. Max Box is a minimum enclosed rectangular area enclosing two boxes, the distance between the diagonals is equal to the maximum distance between the coordinates of the two boxes is defined as c. Among them, the distances are all Euclidean distances, the specific operation steps of AIoU are shown in formulas (2) to (14).

First, the coordinates of anchor and GT are defined:

$$B_{ground} = \left(x_1^g, y_1^g, x_2^g, y_2^g\right) \tag{2}$$

$$B_{anchor} = (x_1^a, y_1^a, x_2^a, y_2^a) \tag{3}$$

For B_{anchor}, make sure that $x_2^a > x_1^a$ and $y_2^a > y_1^a$:

$$\begin{cases} \hat{x}_1^a = min(x_1^a, x_2^a), \hat{x}_2^a = max(x_1^a, x_2^a) \\ \hat{y}_1^a = min(y_1^a, y_2^a), \hat{y}_2^a = max(y_1^a, y_2^a) \end{cases} \tag{4}$$

Forest Chemicals Review
www.forestchemicalsreview.com
ISSN: 1520-0191
July-August 2021 Page No.135-162
Article History: Received: 10 May 2021 Revised: 20 June 2021 Accepted: 18 July 2021 Publication: 31 August 2021

Calculate the area of Anchor and GT:

$$A^{anchor} = (x_2^a - x_1^a) \times (y_2^a - y_1^a) \tag{5}$$

$$A^{ground} = \left(x_2^g - x_1^g\right) \times \left(y_2^g - y_1^g\right) \tag{6}$$

The coordinates of the upper-left corner and the lower-right corner of the outer bounding box can be obtained to obtain c:

$$\begin{cases} out_{max}^x = max\left(\hat{x}_2^a, \hat{x}_2^g\right), out_{max}^y = max\left(\hat{y}_2^a, \hat{y}_2^g\right) \\ out_{min}^x = min\left(\hat{x}_2^a, \hat{x}_2^g\right), out_{min}^y = min\left(\hat{y}_2^a, \hat{y}_2^g\right) \end{cases} \tag{7}$$

$$Diag^{out} = (out_{min}^x)^2 + (out_{min}^y)^2 \tag{8}$$

Calculate A^{in} and d:

$$\begin{cases} in_{max}^x = min(\hat{x}_2^a, \hat{x}_2^g), in_{max}^y = min(\hat{y}_2^a, \hat{y}_2^g) \\ in_{min}^x = min(\hat{x}_2^a, \hat{x}_2^g), in_{min}^y = min(\hat{y}_2^a, \hat{y}_2^g) \end{cases} \tag{9}$$

$$A^{in} = in_{min}^x \times in_{min}^y \tag{10}$$

$$Diag^{A^{in}} = \left(x_2^a - x_1^g\right)^2 + \left(y_2^a - y_1^g\right)^2 \tag{11}$$

Calculate the union of two regions:

$$union = A^{anchor} + A^{ground} - A^{in} \tag{12}$$

Calculate penalties:

$$Pet = \frac{Diag^{in}}{Diag^{out}} \tag{13}$$

Calculate AIoU:

Article History: Received: 10 May 2021 Revised: 20 June 2021 Accepted: 18 July 2021 Publication: 31 August 2021

$$AIoU = \begin{cases} -1, & if \frac{A^{in}}{union} - \frac{Diag^{in}}{Diag^{out}} \leq -1 \\ \frac{A^{in}}{union} - \frac{Diag^{in}}{Diag^{out}}, & if 1 \geq \frac{A^{in}}{union} - \frac{Diag^{in}}{Diag^{out}} > -1 \end{cases} \quad (14)$$

Thus, the regression loss function of Bbox is obtained, as shown in formula (15):

$$L_{AIoU} = 1 - AIoU \quad (15)$$

When the norm is used as loss, it is very sensitive to the scale of the object, resulting in its local optimal value not being the optimal value of IoU [26]. However, the size of the Anchor and the target frame has nothing to do with AIoU, and the loss will not change with the change of the size, because L_{AIoU} is based on a straight-line ratio, not an area ratio. By adding a penalty term to GIoU, the penalty term will be larger, when the distance between the Anchor and the target frame is greater. The gradient direction can also be provided when there is no intersection. The disappearance of the gradient can be limited. In addition, the value range of L_{AIou} is $[-1, 1)$. When the boxes completely overlap, the result tends to 1. When the boxes are far apart, the result is equal to -1, eliminating the problem that calculations cannot be performed without intersection or overlap. The performance comparison is shown in Table I.

Table I. Loss performance comparison

Loss	AP	AP_{50}
L_{IoU}	0.331	0.322
L_{AIoU}	0.393	0.396
L_{GIoU}	0.357	0.381
$smooth - L_1$	0.386	0.387

Through Table I, four Loss functions are used to test the CEPDOF dataset under the Faster R-CNN baseline. When L_{AIoU} is used, it performs better on the detection task. The average precision (AP) is the area under the combined curve of precision and recall, and AP_{50} is the IoU threshold greater than 0.50.

Compared with the other three methods, AIoU is more in line with the regression mechanism. Consider the distance between the object and the anchor point and the coincidence rate so that the object regression returns to stability, and there will be no over-fitting problems during the training process.

3.2 A More Appropriate Activation Function

Article History: Received: 10 May 2021 Revised: 20 June 2021 Accepted: 18 July 2021 Publication: 31 August 2021

In the neural network, Activation functions operate, mapping neuron' input to the output end and introduce nonlinear characteristics so that the network can approximate any nonlinear function arbitrarily.

A smooth activation function allows more practical information to penetrate the neural network, thereby improving the accuracy and generalization ability. The activation function Mish proposed by Misro D, *et al*. [27] is smooth. The positive value can reach any height. Compared with ReLU [28], it avoids the rigid boundary of saturated negative value caused by numerical capping. The slight tolerance of negative value during the transmission process can produce better gradient flow, so this paper chooses to use Mish instead of ReLu. The function comparison is shown in formula (16) and formula (17), and the comparison chart is shown in Fig 3.

$$ReLU = \begin{cases} 0, & if\ x < 0 \\ x, & if\ \ x \geq 0 \end{cases} \tag{16}$$

$$Mish = \begin{cases} 0 & , \quad if\ x = 0 \\ x \times tanh(ln(1 + e^{x})), & if\ x \neq 0 \end{cases} \tag{17}$$

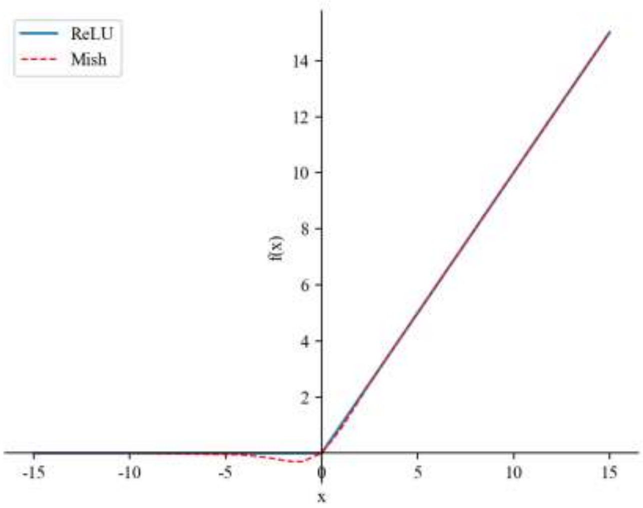


Fig 3. Activation function comparison chart

3.3 Changes in Order between Layers

Article History: Received: 10 May 2021 Revised: 20 June 2021 Accepted: 18 July 2021 Publication: 31 August 2021

The working mechanism of the batch normalization (BN) [29] layer is to help the stochastic gradient descent through the distribution of the hidden layer input. It alleviates the negative impact of the weight update on the subsequent layers during the stochastic gradient descent.

BN and Activation functions appear simultaneously in the existing research, so will the accumulation relationship between the two layers affect the mapping and cause the network performance to decrease? BN is the normalization operation of the input data before the data is transmitted to the network, performed before the input layer. So, from this perspective, BN can be seen as the Normalization of the input passed to the hidden layer. Suppose all the network layers in front of a particular hidden layer in the network are removed. In that case, this hidden layer becomes the input layer. The input passed to it requires Normalization, and the position of this hidden layer is the position of the original BN layer. Therefore, it is natural that the BN layer is placed after the nonlinear activation function.

This article assumes that only the input data is a variable, and the remaining parameters are constants. A comparative experiment is carried out through the derivation of the mathematical relationship between Mish and BN, and the experimental results are shown in Fig 4.

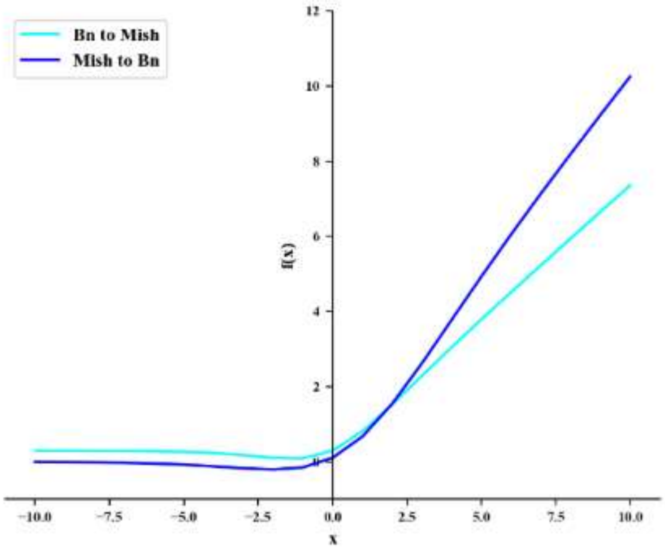


Fig 4. Two different ways to stack the comparison chart

This leads to the conclusion that the Activation function is before the BN layer because the non-negative response of the Activation functions will make the weight update less than ideal [30].

The calculation of the convolutional network is shown in formula (17), and the calculation formula of BN is shown in formula (18).

$$y = \omega \times x + b \tag{18}$$

$$\begin{cases} \mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \\ \sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \\ \hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \end{cases} \tag{19}$$

Through simultaneous formula (17), formula (18), and formula (19), the mathematical relationship of layer stacking can be obtained, as shown in formula (20).

$$\begin{cases} y = \frac{\gamma \times (\omega \times x + b) \times \tanh\left(\ln\left(1 + e^{(\omega \times x + b)}\right)\right)}{\sqrt{\sigma_B^2 + \varepsilon}} - \frac{\gamma \times \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta, & if\ Mish\ to\ BN \\ y = \left(\frac{\gamma \times \omega}{\sqrt{\sigma_B^2 + \varepsilon}} \times x - \frac{\gamma \times \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta\right) \times \tanh\left(\ln\left(1 + e^{\frac{\gamma \times \omega}{\sqrt{\sigma_B^2 + \varepsilon}} \times x - \frac{\gamma \times \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta}\right)\right), & if\ BN\ to\ Mish \end{cases}$$

$$\begin{cases} y = \frac{\gamma \times (\omega \times x + b) \times tanh\left(ln\left(1 + e^{(\omega \times x + b)}\right)\right)}{\sqrt{\sigma_B^2 + \varepsilon}} - \frac{\gamma \times \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta \quad , & if\ Mish\ to\ BN \\ y = \left(\frac{\gamma \times \omega}{\sqrt{\sigma_B^2 + \varepsilon}} \times x - \frac{\gamma \times \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta\right) \times tanh\left(ln\left(1 + e^{\frac{\gamma \times \omega}{\sqrt{\sigma_B^2 + \varepsilon}} \times x - \frac{\gamma \times \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta}\right)\right), & if\ BN\ to\ Mish \end{cases} \tag{20}$$

In the above derivation, m refers to the mini-batch size, μ_B is to calculate the mean, σ_B^2 calculates the standard deviation, $\widehat{x_\iota}$ is the normalization operation, and y_i is the reconstruction transformation.

Assuming that all activation functions are Mish, the convolution value of the negative half is suppressed, and the convolution value of the positive half is retained. When the Mish layer is the BN layer before, ideally, half of the input value is suppressed, and the other half is retained to prevent the gradient from disappearing or exploding. The parameters of the w and BN layers of the convolutional layer are combined to accelerate forward inference.

After BN is used to activate the upper layer, the distribution shape of the nonlinear output will change during training, and limiting its mean (first moment) and variance (second moment) will not eliminate the covariance shift phenomenon. On the contrary, adding BN after the nonlinear activation function will be more symmetrical, non-sparsely distributed, and more Gaussian. The result of the function can be distributed stably.

3.4 Channel Separation Convolution

For the CNN, the score calculation process performs a convolution operation and learns a new feature map from the input feature map through the convolution kernel. Convolution involves extracting spatial (height and weight) and channel features of a local area [31]. In this paper, a unique approach to applying convolution filters, defined as channel separation convolution (CS-conv), is used to augment the convolution layer with additional channel separation, implemented using a different 1D kernel for each channel to do the convolution operation. The composition of the two convolutions constitutes over-parameterization, thereby increasing the learnable parameters, and a single convolution layer can represent the generated linear operation. The convolution process is shown in Fig 5, and the calculation process is shown in the formula (21).

$$\begin{cases} W_i^* = C \times M \times N \times D_k \\ W = \sum_{i=0}^{i} W_i^* \times D_w \times D_h \times M \end{cases} \tag{21}$$

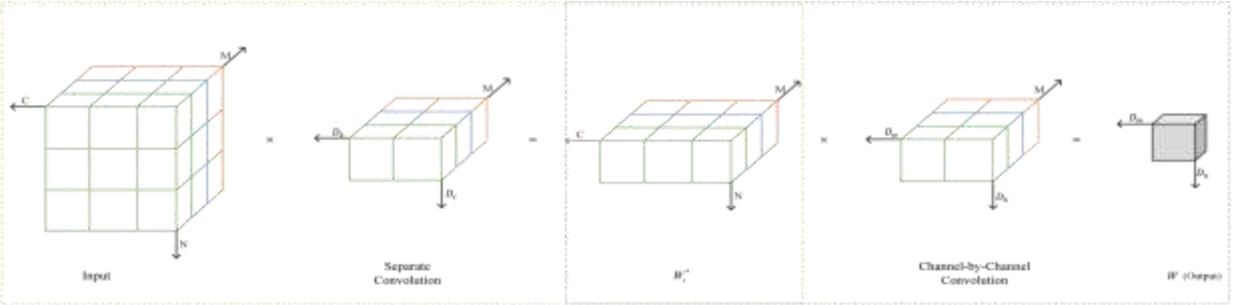


Fig 5. CS-conv convolution process diagram

In formula (21), c represents the number of layers of the feature map, the number of convolutions. In the separate convolution, each channel must be convolved once. M and N represent the dimensions of each channel, and D_k represents the convolution kernel. With the same number of layers as the feature map, the separated convolution w_i^* is obtained, D_w represents the number of layers of the convolution kernel, D_h represents the height dimension of the convolution kernel, and M is the weight dimension. Finally, the channel-by-channel convolution result w is obtained.

The number of parameters is proportional to the size of the convolution kernel. In the separated convolution, there are convolution kernels with a number of M and a size of $D_k \times D_c$, and in the channel-by-channel convolution, there are convolution kernels with a number of M and size of $D_w \times D_h$. The parameters of the whole process are shown in formula (22).

$$D_k \times D_c \times D_w + D_w \times D_h \times M \tag{22}$$

The calculation amount should consider the convolution kernel and the input size: that is, on the basis of the above parameter amount, the input size $D_m \times D_n$ is added to the formula (22), so the calculation amount is as shown in the formula (23):

$$(D_k \times D_c \times D_w + D_w \times D_h \times M) \times (D_m \times D_n) \tag{23}$$

Thus, compared with the standard convolution, the parameter quantity is shown in formula (24), and the calculation quantity is shown in formula (25):

$$\frac{D_k \times D_c \times D_w + D_w \times D_h \times M}{D_k \times D_c \times D_w \times D_h} = \frac{1}{D_h} + \frac{M}{D_k \times D_c} \tag{24}$$

$$\frac{(D_k \times D_c \times D_w + D_w \times D_h \times M) \times (D_m \times D_n)}{(D_k \times D_c \times D_w \times D_h) \times (D_m \times D_n)} = \frac{1}{D_h} + \frac{M}{D_k \times D_c} \tag{25}$$

In the current work in the field of object detection, a 3×3×1 convolution kernel is usually used. The verification formula (24) and formula (25) show that compared with the traditional convolution operation, the amount of calculation and the number of parameters can be reduced by $\frac{1}{3}$, thereby reducing the difficulty of training and avoiding over-fitting problems caused by too many parameters. In summary, CS-conv has a good mathematical conclusion.

ResNet [19] is the idea of adding Residual Learning to the traditional convolutional neural network, which solves the problems of gradient dispersion and accuracy reductions in deep network. Through the description of the above work in this article, the Residual module is improved and defined as Residual_I, as shown in Fig 6.

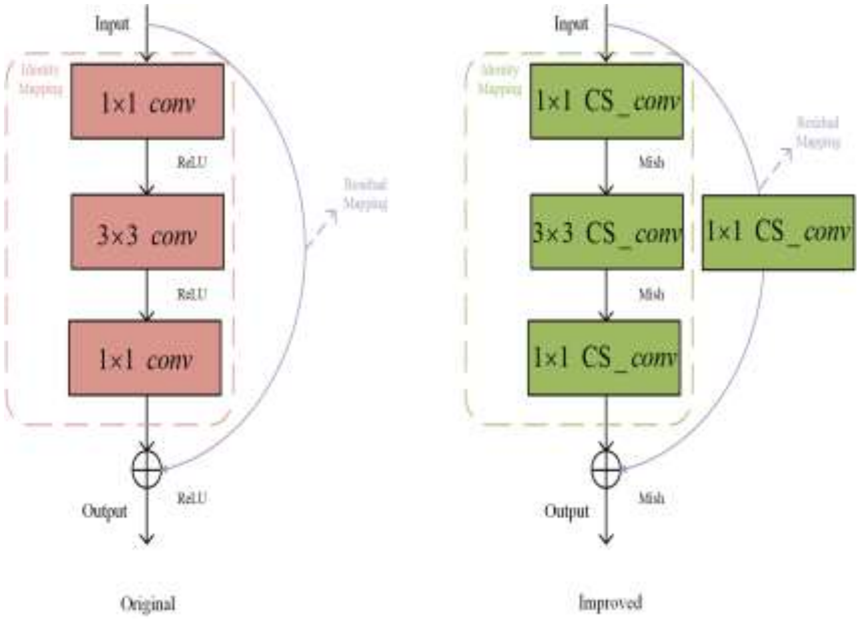


Fig 6. Residual module improvement comparison chart

In improving the Residual module, this paper first replaces the conv layer with the activation function ReLU to CS-conv and Mish. Then, in order to keep the feature map scale constant and to be able to perform linear combinatorial transformations of inter-channel information, CS-conv layers are added to the path of residual mapping. On the CEPDOF data set, the hyper-parameter tuning is not performed. The hyper-parameters tuned for the baseline are used to obtain the comparison results, as shown in Table II.

Table II. Comparison of ResNet and ResNet-I on the CEPDOF dataset

Model	**Depth**	**Combination**	**CEPDOF**
ResNet	50	ReLU-BN-Conv2D BN-ReLU-Conv2D Conv2D-ReLU-BN	0.9351 0.9359 0.9411
		Baseline	0.9291
ReNet-I	50	Mish-BN-CS-conv2D BN-Mish-CS-conv2D CS-conv2D-Mish-BN	0.9438 0.9413 0.9427

		Baseline	0.9230

3.5 Channel Attention Module

The critical research work of this paper is to add the Attention mechanism to the Residual module and achieve a significantly improved detection effect, as shown in Fig 7. The core logic of attention is "from focusing on everything to focusing on key points," focusing limited attention on crucial information, which can save resources and quickly obtain the most effective information. Through this resource allocation mechanism, the initially evenly allocated resources are redistributed according to the importance of attention objects, where the resources to be allocated by attention weight. At present, the attention mechanism is prevalent. The Spatial Transformer Network proposed by Google DeepMind [32], through learning the deformation of the input to complete the processing operation suitable for the task, is a space-based Attention model. Squeeze and excitation net (SENet) [33] is a channel-based Attention model that models the essential characteristics of each channel and then enhances or suppresses different channels for different tasks. There are two ways to realize attention. One is based on reinforcement learning, which is stimulated by the reward function to make the model pay more attention to the details of a specific part. The other is based on gradient descent, which implements the attention mechanism through the objective function and the corresponding optimization function.

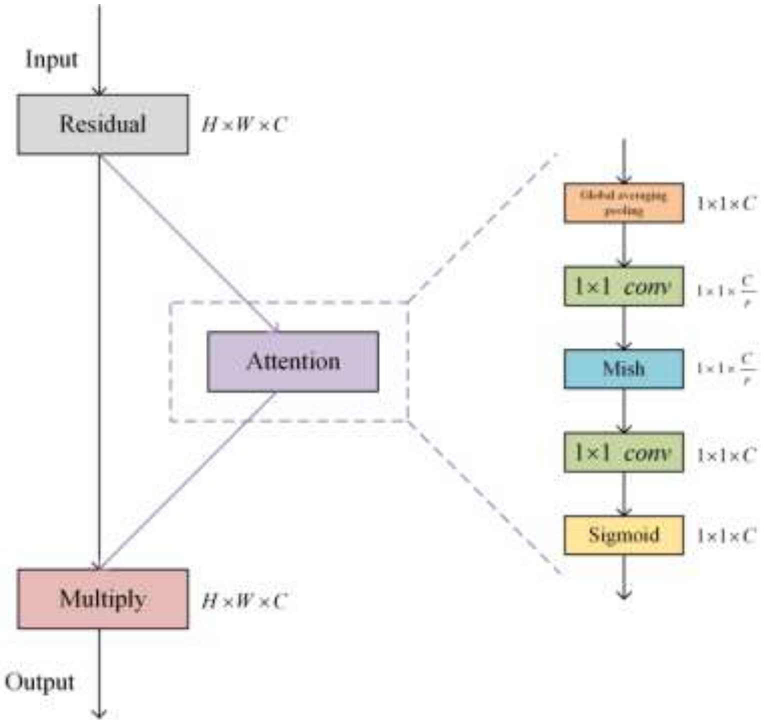


Fig 7. Attention module

The attention module proposed in this article is built based on gradient descent. The central idea is to predict the weight of each output channel and then weight each channel. The first layer of convolution will reduce the number of channels, and the second layer of convolution will increase the number of channels and get the same number of weights as the number of channels. The algorithm design is shown in Table III.

Table III. Attention algorithm table

```
Function：  Attention
Input: channel, r
Output:
    $y \leftarrow Global\ arg\ pool$
    $A \leftarrow channel \times r + \beta$
    $Z \leftarrow \delta(y, A)$
    $B \leftarrow channel \times Z + \beta$
    $S \leftarrow \vartheta(y, B)$
    $X = F_{mu}(channel, S) = S \cdot channel$
End function
```

In the above algorithm table, the $\boldsymbol{Global\ arg\ pool}$ is the global average pooling operation. The $\boldsymbol{\delta}$ represents the Mish activation function, $\boldsymbol{\vartheta}$ is the sigmoid, and the $\boldsymbol{F_{mu}}(,)$ is the dot product function and is defined as $\boldsymbol{F_{mu}(A, B) = B \cdot A}$.

When the attention module is introduced, the number of parameters of the model will be increased. A good trade-off must be made between enhancing performance and improving model complexity. By analyzing the parameters of each layer of ResNet, we choose to use Resnet-50 as the backbone. The analysis process is described below.

Suppose the size of the convolution kernel is $k \times k$, the input channel is M. The output channel is N. When the bias is actual (the number of biases is consistent with the output channel). BN is used (contains two parameters A and B, the parameter quantity is N), and the formula quantity is shown in formula (26). Thus, the backbone parameter comparison table is obtained, as shown in Table IV.

Table IV. ResNet parameter comparison

Model	Depth	Params
ResNet (Basic Block)	10	14356544
	18	33161024
	34	63470656
ResNet (Bottleneck)	50	46159168
	101	85205312
	152	117356032

In Fig 7, r is the reduction factor, which is set to 2, 4, 8, 16, and 32 for experiments. When r is equal to 16, the effect is the best, as shown in Table V.

Table V. Reduction factor comparison

Ratio(r)	Acc	Params
2	0.7671	45.7M
4	0.7675	35.7M
8	0.7674	30.7M
16	0.7682	28.1M
32	0.7628	26.9M
Original	0.7571	25.6M

In the above table, Acc is the accuracy rate in CEPDOF, params are the model parameter amount, and the original represents ResNet-50 without the attention mechanism.

Based on existing research, this paper replaces the 7×7 convolutional layer with three 3×3 convolutional layers. The advantage of this is to reduce the number of parameters and increase the non-linear ability. The effect of stacking multiple convolutional layers lies in the extraction and recombination of features. The extracted features range from simple edges to contours to more complex high-level features. Another point, the convolution operation does not destroy the spatial information of the image. The fully connected layer is abstracted into a convolutional layer with a large convolution kernel. The convolutional layer is abstracted into a sparse, fully connected layer, converting the final complete connect (FC) layer into a 1×1 convolutional layer. This operation has two advantages: first, in terms of spatial information, the FC layer converts the data into a one-dimensional vector. Although there are pixel characteristics, the spatial position information is severely lost. The conv layer compresses it into a thumbnail to save more features; secondly, the convolution operation will not destroy the end-to-end network. In summary, this article defines the upgraded ResNet as ResNet_I, and the network structure is shown in Table VI.

Table VI. ResNet-I network structure table

Layer_name	**Output_size**	**Attention-ResNet-50**
Conv1	112×112	CS-Conv,3@3×3,stride 2
Conv2_x	56×56	$\begin{pmatrix} \mathrm{CS-Conv}, & 1\times 1, & 64 \\ \mathrm{CS-Conv}, & 3\times 3, & 64 \\ \mathrm{CS-Conv}, & 1\times 1, & 256 \\ f_A, & & [16{,}256] \end{pmatrix} \times 3$
Conv3_x	28×28	$\begin{pmatrix} \mathrm{CS-Conv}, & 1\times 1, & 128 \\ \mathrm{CS-Conv}, & 3\times 3, & 128 \\ \mathrm{CS-Conv}, & 1\times 1, & 512 \\ f_A, & & [32{,}512] \end{pmatrix} \times 4$
Conv4_x	14×14	$\begin{pmatrix} \mathrm{CS-Conv}, & 1\times 1, & 256 \\ \mathrm{CS-Conv}, & 3\times 3, & 256 \\ \mathrm{CS-Conv}, & 1\times 1, & 1024 \\ f_A, & & [64{,}1024] \end{pmatrix} \times 6$
Conv5_x	7×7	$\begin{pmatrix} \mathrm{CS-Conv}, & 1\times 1, & 512 \\ \mathrm{CS-Conv}, & 3\times 3, & 512 \\ \mathrm{CS-Conv}, & 1\times 1, & 2048 \\ f_A, & & [128{,}2048] \end{pmatrix} \times 3$

IV. THE ALGORITHM AND ITS RESULTS

4.1 Algorithm Interpretation of This Paper

This paper adopts a two-stage detection algorithm that combines positioning and recognition and designs a person detection algorithm under a fisheye lens based on deep learning to support full-image end-to-end training. The algorithm framework is shown in Fig 8.

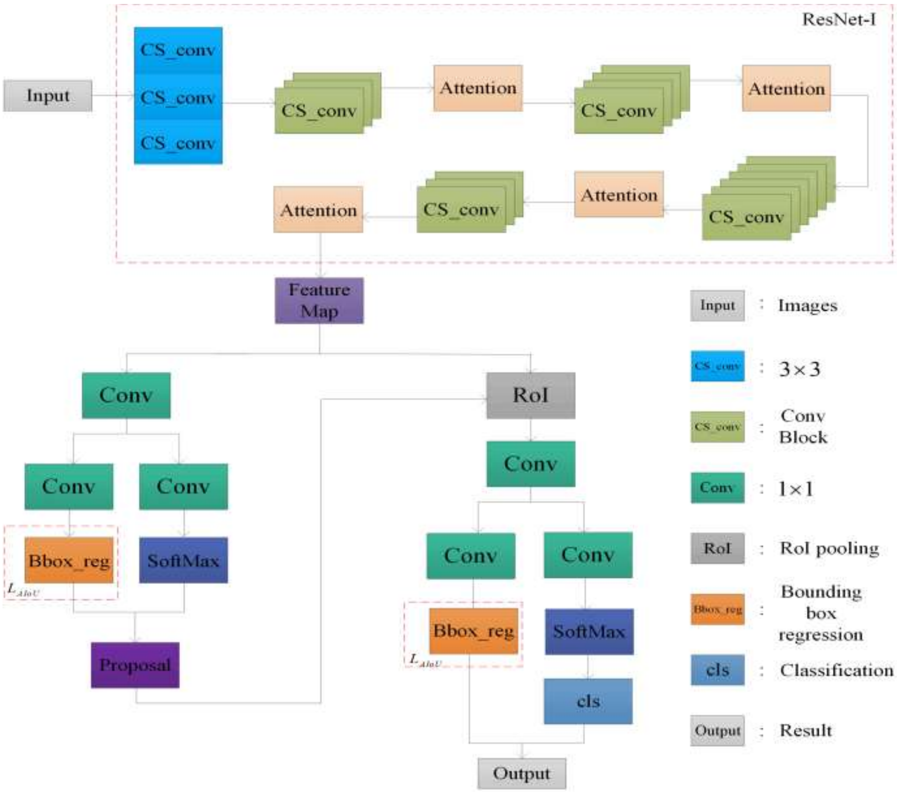


Fig 8. FED structure diagram

● Use ResNet-I as Backbone to extract Feature Map, increase Feature saliency through CS-conv and Attention, and change layer stacking to accelerate forward inference and improve network accuracy.

● In order to detect the target more accurately, the feature map is transmitted to the positioning network, and a proposal is generated. The AIoU proposed in this paper is used for bounding box regression, and the Softmax is used to determine whether there is a target object.

● Combine the region proposal generated by the proposal with the Feature Map, use ROI Pooling for pooling to obtain a fixed region proposal, and enter the subsequent bounding box regression module through the convolutional layer to perform boundary correction and SoftMax category classification.

● A fully convolutional network is realized, which supports input of different sizes, can better learn context information, and is suitable for image tasks.

4.2 Dataset

Article History: Received: 10 May 2021 Revised: 20 June 2021 Accepted: 18 July 2021 Publication: 31 August 2021

Some existing data sets are used to detect people under indoor fisheye lenses. On the one hand, no overhead lens is used, and on the other hand, the number of frames and the number of people is limited [34]. In order to solve the problem and test the accuracy of the algorithm proposed in this paper in the detection of people under the fisheye lens indoors, this paper chose the CEPDOF data set released by the Visual Information Processing Laboratory of Boston University in April 2020. This data set is a small classroom recorded by an overhead fisheye camera. Up to 13 people can be seen at a time, with 25,504 annotated frames. The scenes inside are very challenging, such as crowded rooms and body occlusion, Human movement distortion and projection, head camouflage (wear a hat or mask). First, divide the data set into a training set and a test set with a ratio of 0.8:0.2; then, divide the training set into a sub-training set and a validation set with a ratio of 0.75:0.25; finally get the sub-training set, validation set, and test set. The ratio between the three groups is 6:2:2. The training set is used to obtain the weights and parameters of the network. The verification set is used to verify the effect adjustment parameters. In the training process, the model is used to predict the validation set. The parameters are fine-tuned according to the validation set results, and the parameters and activation functions corresponding to the optimal model are selected. The test set is used to test the detection accuracy and generalization performance of the model.

4.3 Performance Metrics

The evaluation standard of this model follows MS COCO and adopts AP as one of the evaluation standards, outcome classification as shown in Table VII. This article only considers AP when AIoU = 0.5(AP_{50}) because even a very complete person detection algorithm may have a relatively low IoU, which is caused by the non-uniqueness of GT. Different but equally effective bounding boxes will be produced at different angles for the same person under the same lens. However, only one label will be annotated as GT by the inspector. In addition to AP, F-measure and Accuracy with fixed confidence $b_{conf} = 0.3$ and $\alpha = 1$ are also used as another evaluation index.

Table VII. Outcome classification confusion matrix

The true situation	**Forecast result**	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

The reason for using F-measure in the evaluation standard is that there will be conflicts

Article History: Received: 10 May 2021 Revised: 20 June 2021 Accepted: 18 July 2021 Publication: 31 August 2021

between accuracy (P) and recall (R). At the same time, F-measure is a weighted and harmonic evaluation of $P$and$R$, and α is the weight. When $\alpha = 1$, F -Measure combines the results of P and R. When the F-measure result is higher, it can indicate that the method is more effective than formula (26) and formula (27).

$$\begin{cases} P = \frac{TP}{TP+FP} \\ R = \frac{TP}{TP+FN} \end{cases} \tag{26}$$

$$F - Measure_{\alpha=1} = \frac{(\alpha^2+1)\times P\times R}{\alpha^2\times(P+R)} = \frac{2\times P\times R}{P+R} \tag{27}$$

Accuracy represents the proportion of correctly detected samples in the total number of detected samples, and R represents the proportion of correctly detected samples in the total detected samples. The calculation method of Accuracy is shown in formula (28).

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \tag{28}$$

4.4 Implement Details

The experimental environment is as follows: Intel Xeon E5-2682v4@2.5GHz, 8VCPU; memory (RAM): 128 GB; discrete graphics card: GeForce GTX T4; system type: 64-bit Ubuntu 18.04; development tools: Python 3.6, based on Pytorch's mmdetection framework [35].

The method of conducting the simulation experiment is as follows. First, train the network on MS COCO 2017, train the images for 10,000 iterations, and then perform 800 iterations of fine-tuning on CEPDOF, with a maximum of 128 images in one iteration. In the COCO image, the network weight is updated by SGD [36] and has the following parameters: step size 0.001, momentum 0.9, weight 0.0005. For the CEDPOF data set, the standard SGD step size is 0.0001. Albumentation [37] is applied in the training phase. In order to prevent overfitting, dropout [38] is used, and the random deletion probability is set to 0.5. All results are based on only one training and inference.

4.5 Comparative Experiment and Analysis

In the comparison experiment, the CEPDOF data set was used for comparison in the same experimental environment. First, the Faster R-CNN baseline was compared, and then ResNet-I, Res2Net [39], and OHEM [40] were migrated to the baseline using fine-tuning in migration learning. In addition, compared with DPM [7], RAPID [5], and Fovea [41], FED has a better detection effect at this stage. In the experimental results, after 800 iterations of the model, FED shows the highest value of acc and the lowest value of the loss. When the two curves are jointly observed, there is no over-fitting phenomenon, and the model has good generalization ability. Compared with other models in the line chart, the acc of FED is stable, and the loss is smooth and stable, as shown in Fig 9. Then, use the performance metrics described in this article for evaluation, as shown in Table VIII. Compared with other models, FED better detects people under the fisheye lens when looking down.

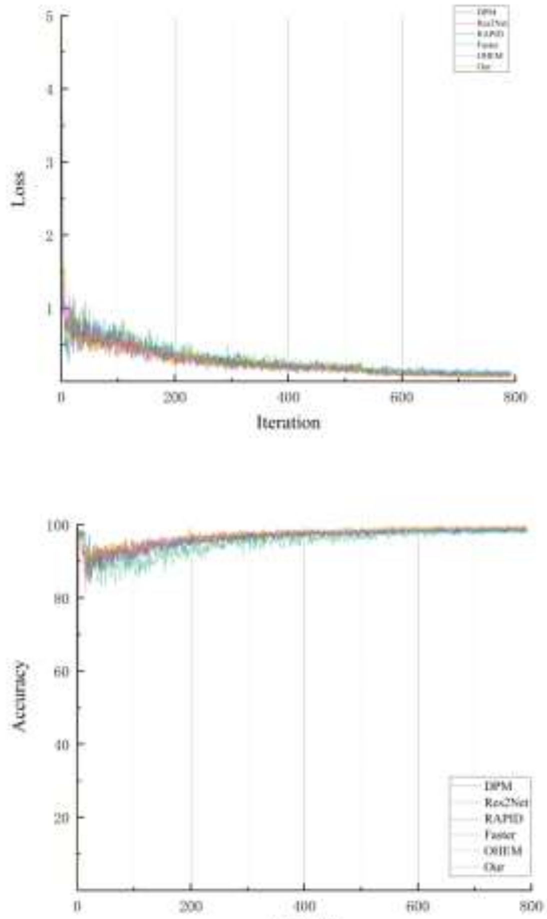


Fig 9. Accuracy and Loss comparison chart

Table VIII. Comparison results of different models on CEPDOF dataset

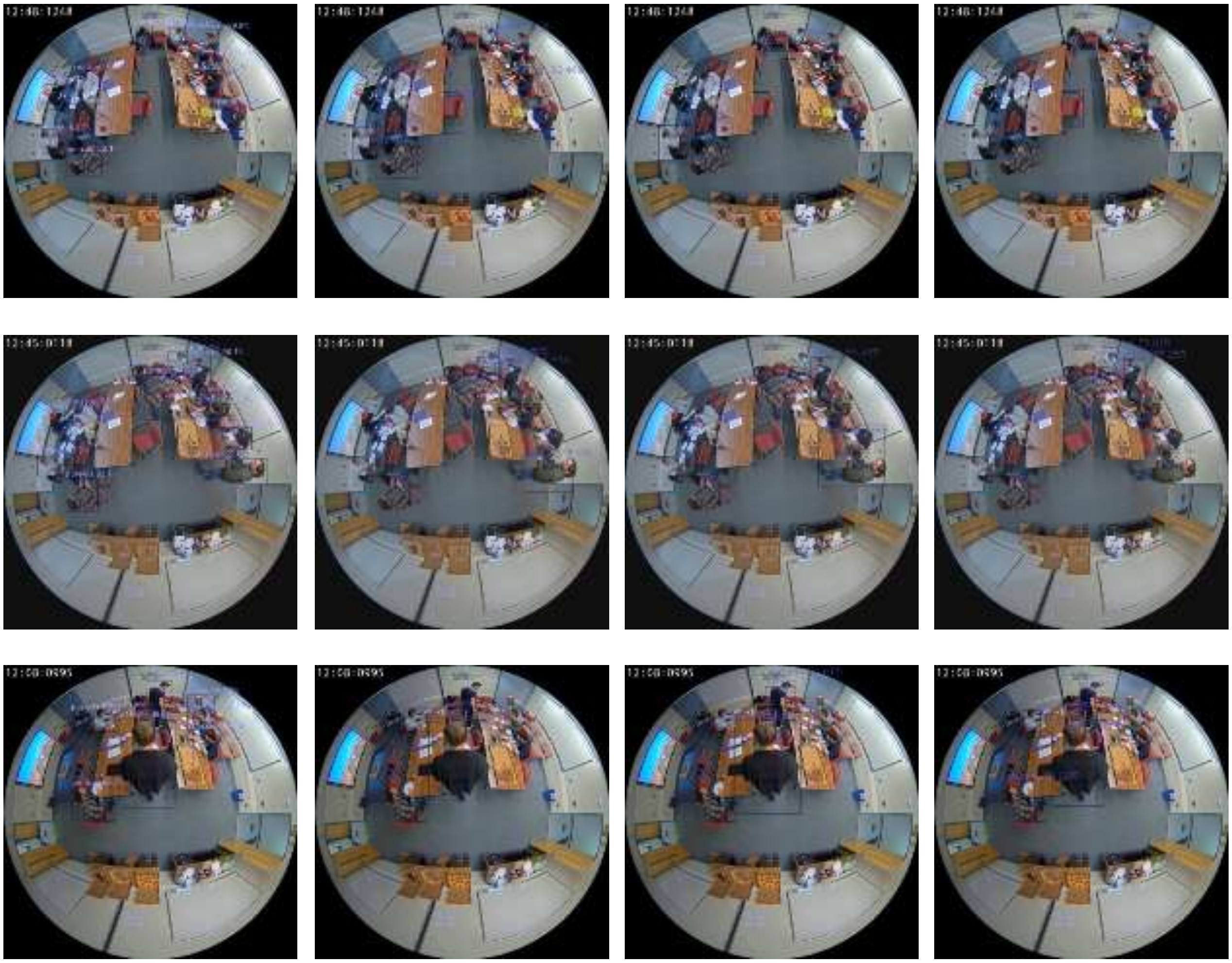
Model	**CEPDOF**					
	FPS	AP_{50}	Acc	Recall	F1	Inference
DPM	6.8	73.9	0.896	0.638	0.683	71
Faster	5.1	75.7	0.904	0.656	0.691	78
RAPID	7.0	82.4	0.911	0.719	0.793	57
OHEM	5.9	77.2	0.902	0.696	0.744	73
Res2Net	4.6	81.5	0.912	0.731	0.798	69
Fovea	4.9	85.1	0.907	0.816	0.831	83
Our	6.7	85.3	0.953	0.847	0.834	62

In Table VIII, FPS is the frame rate per second, that is, the number of images processed per second (evaluated in the same environment). In the comparison of FPS, Res2Net is the fastest, but other indicators are too far away from FED; in the comparison of Recall and Acc, not only FED increases the inference speed, but Recall and Acc are in a leading position; in the comparison of inference, RAPID is the fastest, and other indicators are not competitive with FED. The result is obtained by averaging the scores of the data set.

Through the above analysis, in this field, most of the FED evaluation indicators are in the leading position, indicating that this model is more competitive than other models and the results obtained are better.

4.6 Effect Picture

This section uses the FED structure on GeForce GTX T4 to perform random image comparison tests (for personnel detection only) against three models: DPM, RAPID, and Fovea. For ease of observation, we provide four sets of comparison results. The FED is a leader in comparing personnel positioning and recognition accuracy, as shown in Fig 10.



(a) Our (b) DPM (c) RAPID (d) Fovea

Fig 10. The person comparison results graph, column (a) is the FED presented in this article, (b) is DPM, (c) is RAPID, and (d) is Fovea group per line.

V. CONCLUSION

This paper proposes a model FED based on the Faster R-CNN architecture for the real-life people detection scene looking down under the fisheye lens. Faced with practical requirements, background interference, and non-upright state challenges, L_{AIoU} is used to locate and solve. Since L_n cannot accurately reflect the overlapping relationship between anchor and GT, using a channel-based Attention module can help the model choose better features. FED's performance is better than existing algorithms for the specified data set without introducing additional computational complexity. This algorithm shows good performance detecting people under the fisheye lens, making it a reference value in the field of intelligent security and has a strong application prospect in civil or military applications. In addition, this research still has shortcomings. Due to the impact of lens, pixels, lighting conditions, its actual application effect may differ from the experimental results, and the scale of the experiment needs to be expanded.

ACKNOWLEDGEMENTS

This work is supported under the key research and development plan of China's Gansu Province-Industrial Category: UAV Key Technology Research (Project Number: 18YF1GA060).

REFERENCES

[1] D. Hossain, G. Capi and M. Jindai, “Object recognition and robot grasping: A deep learning based approach,” in RSJ, Yamagata, Japan, pp. 1-5, 2016.

[2] X. Yin, X. Wang, J. Yu, et al., “Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification,” in ECCV, Munich, Germany, pp. 469-484, 2018.

[3] A. Krizhevsky, I. Sutskever, G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, vol. 25, pp. 1097-1105. Dec. 2012.

[4] S. Li, M. O. Tezcan, P. Ishwar, et al., “Supervised people counting using an overhead fisheye camera,” in AVSS, Guangzhou, China, pp. 1-8, 2019.

[5] Z. Duan, O. Tezcan, H. Nakamura, et al., “RAPiD: rotation-aware people detection in overhead fisheye images,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. Seattle, Washington, USA, pp. 636-637, 2020.

[6] S. Ren, K. He, R. Girshick, et al., “Faster R-CNN: towards real-time object detection with region proposal networks,” IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 6, pp. 1137-1149, Jun. 2016.

[7] Z. Zou, Z. Shi, Y. Guo, et al., “Object detection in 20 years: A survey,” arXiv preprint arXiv:1905.05055, 2019.

[8] J. Long, E. Shelhamer, T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, Boston, Massachusetts, USA, pp. 3431-3440, 2015.

[9] A. Palla, D. Moloney, L. Fanucci, “Fully Convolutional Denoising Autoencoder for 3D Scene Reconstruction from a single depth image,” in 4th ICSAI, shanghai, China, pp. 566-575, 2017.

[10] J. Mas, T. Panadero, G. Botella, et al., “CNN Inference acceleration using low-power devices for human monitoring and security scenarios,” Computers & Electrical Engineering, vol. 88, pp. 15-21, Apr. 2020.

[11] D. Navneet, B. Triggs, “Histograms of oriented gradients for human detection,” in Proc. IEEE Conf. Computer Vision Pattern Recognition, Boston, Massachusetts, USA, pp. 886–893, 2015.

[12] L. Bastian, E. Seemann, B. Schiele, “Pedestrian detection in crowded scenes,” in Proc. IEEE Conf. Computer Vision Pattern Recognition, San Diego, Calif, USA, pp. 716–723, 2005.

[13] W. Liu, D. Anguelov, D. Erhan, et al., “SSD: Single Shot MultiBox Detector,” in Proceedings of

European Conference on Computer Vision, Berlin, Amsterdam, Netherlands, pp.21-37, 2016.

[14] J. Redmon, S. Divvala, R Girshick, et al., “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, USA, pp. 779-788, 2016.

[15] W. Xia, Z. Wei, “Multi-view indoor personnel detection network based on joint learning,” Acta Optics, vol. 39, no. 2, pp.78-88, Nov, 2019.

[16] K. Simonyan, A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” Computer Science, 2014.

[17] K. He, X. Zhang, S. Ren, et al., “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, USA, pp. 770-778, 2016.

[18] A. T. Chiang, Y. Wang, “Human detection in fish-eye images using HOG-based detectors over rotated windows,” In ICMEW, Chengdu, China, pp. 1-6, 2014.

[19] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, et al., “Object detection with discriminatively trained part-based models,” IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 9, pp.1627-1645, Sep, 2009.

[20] B. Yang, J. Yan, Z. Lei, et al., “Aggregate channel features for multi-view face detection,” in IEEE international joint conference on biometrics, Tampa, Florida, USA, pp.1-8, 2014.

[21] M. Tamura, S. Horiguchi, T. Murakami, “Omnidirectional pedestrian detection by rotation invariant training,” in Proc. of IEEE Winter Conf. on Applications of Computer Vision, Snowmass Village, Colorado, USA, pp. 1989–1998, 2019.

[22] T. Y. Lin, M. Maire, S. Belongie, et al., “Microsoft coco: Common objects in context,” in European conference on computer vision, Zurich, Switzerlan, pp. 740-755, 2014.

[23] Y. Xu, M. Fu, Q. Wang, et al., “Gliding vertex on the horizontal bounding box for multi-oriented object detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 4, pp. 1452-1459, Apr, 2020.

[24] X. Shu, D. Yuan, Q. Liu, et al., “Adaptive weight part-based convolutional network for person re-identification,” Multimedia Tools and Applications, vol.79, no.31, pp. 23617-23632, Aug, 2020.

[25] H. Rezatofighi, N. Tsoi, J. Y. Gwak, et al., “Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression,” in CVPR, Seattle, USA, pp. 658-666, 2020.

[26] B. Jiang, R. Luo, J. Mao, et al., “Acquisition of localization confidence for accurate object detection,” in ECCV, Munich Germany, pp. 784-799, 2018.

[27] D. Misra, “Mish: A self-regularized non-monotonic activation function,” unpublished.

[28] X. Glorot, A. Bordes, Y. Bengio, “Deep sparse rectifier neural networks,” in Proceedings of the fourteenth international conference on artificial intelligence and statistics, Nanjing, China, pp. 315-323, 2011.

[29] S. Ioffe, C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” In International conference on machine learning, Lile, France, pp. 448-456, 2015.

[30] G. Chen, P. Chen, Y. Shi, et al., “Rethinking the usage of batch normalization and dropout in the training of deep neural networks,” unpublished.

[31] L. Liu, W. Ouyang, X. Wang, et al., “Deep learning for generic object detection: A survey,” International journal of computer vision, vol. 128, no.2, pp. 261-318, Jul, 2020.

[32] M. Jaderberg, K. Simonyan, A. Zisserman. “Spatial transformer networks,” Advances in neural

Article History: Received: 10 May 2021 Revised: 20 June 2021 Accepted: 18 July 2021 Publication: 31 August 2021

information processing systems, vol.28, no.1, pp. 2017-2025, Mar, 2015.

[33] J. Hu, L. Shen, G. Sun. “Squeeze-and-excitation networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, USA, pp.7132-7141, 2018.

[34] G. Qun, Z. Jun, W. Qianqian, C. Jie, X. Chao, “Research on target detection algorithm based on fisheye image,” Control and Information Technology, vol. 12, no. 3, pp. 43-47, Mar, 2019.

[35] K. Chen, J. Wang, J. Pang, et al., “Mmdetection: Open mmlab detection toolbox and benchmark,” unpublished.

[36] J. M. Cherry, C. Adler, C. Ball, et al., “SGD: Saccharomyces genome database,” Nucleic acids research, vol. 26, no. 1, pp. 73-79, Jan, 1998.

[37] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, et al., “Albumentations: fast and flexible image augmentations,” Information, vol. 11, no. 2, pp. 125-136, Feb, 2020.

[38] N. Srivastava, G. Hinton, A. Krizhevsky, et al., “Dropout: a simple way to prevent neural networks from overfitting,” The journal of machine learning research, vol.15, no.1, pp. 1929-1958, May, 2014.

[39] S. Gao, M. Cheng, K. Zhao, et al., “Res2Net: A New Multi-scale Backbone Architecture,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no.2, pp. 652-662, Jun, 2019.

[40] A. Shrivastava, A. Gupta, R. Girshick, “Training region-based object detectors with online hard example mining,” in Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, USA, pp. 761-769, 2016.

[41] T. Kong, F. Sun F, H. Liu, et al., “Foveabox: Beyound anchor-based object detection,” IEEE Transactions on Image Processing, vol. 29, no. 2, pp. 7389-7398, Sep, 2020.