Fast EfficientDet: An Efficient Pedestrian Detection Network

M. Y. Cao, J. Zhao

***Abstract*—As an essential application in object detection, pedestrian detection has received extensive attention in many areas such as autonomous driving, video surveillance, and criminal investigation. With the rapid development of deep learning, pedestrian detection has made significant progress. When faced with multi-scale target pedestrians and dense crowds, false and missed detections are prone to occur, affecting accuracy. To overcome this problem, this study presents a multi-scale object detection network (Fast EfficientDet), based on an improved EfficientDet. Firstly, the backbone network EfficientNet is improved, and some of the deepwise separable convolutions that affect the speed of the model in the early training stage are discarded. At the same time, the Mish activation function is introduced to speed up the model's training. Secondly, a new feature pyramid-network Skip-BiFPN is proposed. Based on BiFPN, a cross-layer data stream is added to integrate the object's semantic and location information. In the face of complex environments, the network can better detect objects with large differences in size. Finally, the DIoU calculation method is introduced in the NMS post-processing. The suppression problem between the candidate frames is better handled by referring to the center point distance to solve object occlusion. Compared to the original EfficientDet series algorithm, the Fast EfficientDet-D0 obtained the best mAP of 84.98%, and the training speed increased by 15%. Compared to other algorithms, the Fast EfficientDet model has better performance.**

***Index Terms*—VOC 2012, EfficientNet, Object detection, EfficientDet.**

I. Introduction

In recent years, several human-related research directions have arisen in the field of computer vision. These include pedestrian detection, pedestrian re-identification, action recognition, human pose estimation, face recognition, and many more. As one of the important research directions in computer vision, pedestrian detection has been a hot research topic due to its high practicality, including video surveillance, drones, and autonomous vehicles, and is an essential task in intelligent surveillance systems. Its primary objective is detecting and localization of all pedestrians in each picture frame in a given scene. When monitoring open areas, such as shopping malls, schools, and road scenes, pedestrians are at different distances from each other, and the size of the pedestrians captured by the surveillance varies, thus creating multi-scale objects. In highly crowded and complex environments, pedestrians are subject to severe background interference and occlusion from similar pedestrian objects and other objects, making them susceptible to missed and false detections. General pedestrian detectors are not well suited to solve this type of problem.

The key to pedestrian detection lies in extracting features. Most of the traditional methods for extracting features are manual, such as HOG [1], Haar [2], and LBP [3]. In the actual complex context, the features extracted by such traditional methods are hardly robust, so the results are not ideal. Later, convolutional neural networks (CNN), such as AlexNet [4] and VGGNet [5], became widely used in object detection, as they can extract more complex features and improve detection accuracy. R-CNN [6] is the starting point of deep learning models in pedestrian detection. Using a sliding window, this two-stage object-detector extracts features. The subsequent Faster R-CNN [7] generated proposals utilizing RPN, significantly improving performance and becoming a popular detection framework for pedestrian detection. Later, the one-stage object detectors were proposed, such as the YOLO [8]-[11] and the SSD [12]-[14] series of algorithms, and RetinaNet [15], which perform classification and regression simultaneously, much faster than the two-stage model but with lower accuracy. One of the representatives, SSD, directly classifies and regresses the anchor points, and it can directly get the bounding box. The detection results of the existing deep learning-based detection algorithms are closely related to the features extracted from the available CNN. How these features are fully utilized from low-level positional information to high-level semantic information directly determines the model's performance. The rational use of feature maps at different scales becomes a top priority for pedestrian detection tasks.

This study proposes a pedestrian detection algorithm called Fast-EfficientDet. Compared with EfficientDet, Fast-EfficientDet has better detection accuracy and faster training speed. The main contributions of this paper are three-fold: (1) We propose a new backbone network, EfficientNet+, where the ordinary 3×3 convolutions are used to replace the deep separable convolution in the MBConv and using the Mish activation function improves the training speed of the model. (2) We propose a new feature pyramid network called Skip-BiFPN. Based on BiFPN, cross-layer data flow is added to fully use the semantic relationships and

Manuscript received August 26, 2021; revised February 08, 2022. This work was supported by the Natural Science Foundation of Liaoning Province, under Grant 20180551048.

M. Y. Cao is a Master's Student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China (e-mail: a360239748@163.com).

J. Zhao is a Professor of School of Computer Science and Software Engineering, University of Science and Technology, Liaoning, Anshan 114051, China (corresponding author, phone: 86-13998086167; e-mail: zhaoji@ustl.edu.cn).

location information between the upper and lower layers to achieve multi-layer and multi-node fusion learning. (3) We introduce the DIoU calculation in the traditional NMS algorithm, using the distance between the centroids of the objects as a criterion to effectively solve the severe occlusion problem in pedestrian detection.

The remainder of this paper is organized as follows: Section II describes the current national and international related work on pedestrian detection. Section III introduces EfficientDet and EfficientNet networks. Section IV describes our proposed EfficientNet+, Skip-BiFPN, and DIoU-NMS in detail. Section 5 describes our experimental procedure and discusses the experimental results in detail. Conclusions are presented in Section VI.

II. Related Work

The development of pedestrian detection can be divided into two phases. The first is using manually constructed object features and the corresponding classification algorithm to obtain the detection results. This type of approach is referred to as "feature extractor + classifier." Prior to introducting convolutional neural networks, the standard approach to pedestrian detection was to extract manually constructed features from all locations within a sliding window. One of the most classical methods is the HOG+SVM-based pedestrian detection algorithm proposed by Dalal and Triggs [16]. Pedestrian detection is done by calculating and counting the gradient change information of the local area of the image as a feature of the picture. By using a linear classifier for the detection of objects, the effect of small deformations on the object to be detected can be effectively eliminated. However, this method is not ideal for detection against complex backgrounds and occlusions. To address the object occlusion problem, P. F. Felzenszwalb et al.[17] proposed the Deformable Component Model (DPM), a further extension of the HOG capable of detecting pedestrians in different poses using variable components and distinguishing between objects and background. P. Dollar et al.[18] proposed an integrated channel feature (ICF) algorithm for hybrid features, using the integral graph technique to compute each feature channel in an image to efficiently fuse the relevant pedestrian features. These earlier works used mostly SVM or Random Forest classifiers, the model takes time for feature computation, and the detection time of the classifier was too long, affecting the real-time performance.

The second stage is to use convolutional neural networks to extract features. In recent years, deep learning-based methods have been proposed, CNNs have become the main object detection method, and researchers have chosen to apply deep learning to pedestrian detection. Zhang et al.[19] analyzed the poor performance of Fast-RCNN in pedestrian detection, proposing Region Proposal Network (RPN) to generate candidate regions directly and boost Forest for classification. Based on Fast-RCNN, Li et al.[20] used two sub-networks to detect pedestrians of different scales and employed a scale-aware weighting mechanism to reduce the effect of object scale on detection accuracy. Yang et al. [21] proposed a deep neural network fusion architecture in which the SSD network first generates all possible pedestrian candidates of different sizes and occlusion levels and further refines the pedestrian candidates. This approach works well in detecting small-sized and occluded pedestrians. Mendes et al.[22] proposed a unified deep neural network for fast detection of multi-scale objects. Detection is performed at different intermediate network layers, and multi-scale feature maps match pedestrians of different scales. The method saves memory and computation significantly and achieves high detection rates up to 15 fps. Guan et al.[23] proposed an unsupervised way where the 3D pedestrian virtual space is constructed based on detection and tracking for only one pedestrian, and the pedestrian motion pattern is mapped into a 3D virtual space instead of a traditional 2D image space. The proposed approach is efficiently distinguishes the anomaly without any hypothesis for the scenario contents in advance.

Despite the progress made by these methods, there is a continued need to explore new ideas and approaches. As pedestrian detection plays an increasingly important role in our lives, increasing research is dedicated to dealing with more complex scenarios. Essentially, all these networks are looking to extract features by designing deeper networks. This study aims to make fuller use of multi-level features and performs a multi-scale fusion.

III. The Model Structure Of EfficientDet

Tan et al. [24] proposed EfficientDet, an object detector that balances model accuracy and detection speed simultaneously. The model structure shown in Fig. 1 consists of the backbone network EfficientNet, the bi-directional feature extraction network BiFPN and the box/class prediction net.

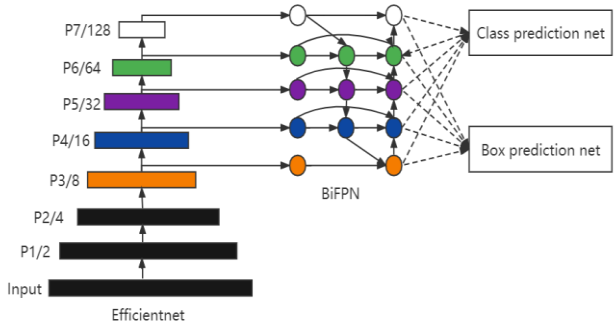

Fig. 1. EfficientDet network structure.

When talking about EfficientDet, it is essential first to introduce its backbone network, EfficientNet. EfficientNet [25] is a classification network proposed in 2019. To improve the accuracy of the network, EfficientNet proposes scaling. It starts by increasing the width of the baseline network, using more convolutional kernels for each convolutional layer, boosting the number of feature matrix channels. The depth was then added to the baseline network, which means more layers in the network. Next, the resolution of the input image was increased on the baseline network, and the height and width of each feature matrix were increased accordingly. Finally, the width, depth, and resolution were boosted simultaneously on the baseline network. When scaling is combined, the performance is better for the same amount of FLOPs. It is demonstrated that the best scaling method is obtained by combining multiple dimensions. A combination factor represents the change in computational resources,

controlling how many resources are available for model scaling. The depth is scaled according to α^{ϕ}, the width according to β^{ϕ}, and the resolution according to γ^{ϕ}. The constraint $\alpha \cdot \beta^2 \bullet \gamma^2 \approx 2$. When the computational resources differ, it is sufficient to calculate what ϕ is to know the corresponding width, depth, and resolution.

The benchmark network EfficientNet-B0 was obtained as the baseline model of the EfficientNet family by searching through the NAS neural architecture. By combining scaling on such a basis, eight versions of the EfficientNet network were explored, EfficientNet-B0 to B7. BiFPN is based on the idea of multi-scale feature fusion, which is improved on PANet. Cross-scale connection optimization methods obtain more feature fusion. When fusing features of different resolutions, EfficientDet proposes a weighted fusion method for each input feature, called fast normalized fusion, and lets the network learn the weights of each input. EfficientDet inherits the idea of EfficientNet by combining the scaling of the model while combining EfficientNet, BiFPN, and the box/class prediction net are hybrid scaled to obtain eight versions, namely EfficientDet-D0 to D7.

Real-world environments are subject to interference from external situations such as occlusion and multiple scales. The direct use of EfficientDet for pedestrian detection has some shortcomings. For multi-scale objects, there is a lack of shallow feature extraction. When an obscuring situation occurs, EfficientDet does not perform well. Moreover, although the number of parameters in EfficientDet is small, the number of layers in the network is too high. The backbone network uses much deepwise separable convolution, requiring too much video memory during training, which directly affects the training time of the network. Therefore, the training time of the network is directly affected. To solve these problems, we optimize and improve the model based on EfficientDet.

IV. Fast EfficientDet

The proposed main structural improvements based on EfficientDet is as follows: The ordinary 3×3 convolution replaces the deep separable convolution in the MBConv. In Neck BiFPN, cross-layer data flow and fusion of more nodes are presented for better object feature extraction. The improved network was named Fast EfficientDet. The overall Fast EfficientDet network structure is shown in Fig. 2.

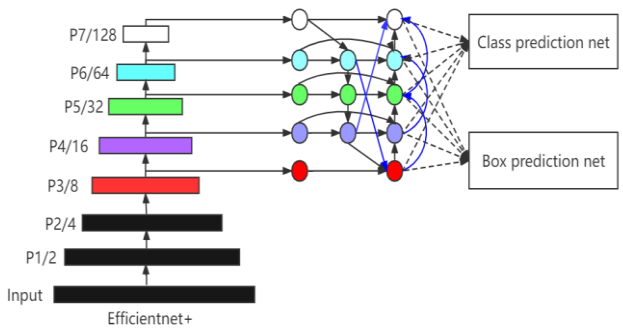

Fig. 2. Fast EfficientDet network structure.

A. Backbone Network EfficientNet+

TABLE I
EfficientNet-B0 Architecture

Stage i	Operator F_i	Resolution $H_i \times W_i$	Channels F_i	Layers L_i
1	Conv3×3	224×224	32	1
2	MBConv1，k3×3	112×112	16	1
3	MBConv6，k3×3	112×112	24	2
4	MBConv6，k5×5	56×56	40	2
5	MBConv6，k3×3	28×28	80	3
6	MBConv6，k5×5	14×14	112	3
7	MBConv6，k5×5	14×14	192	4
8	MBConv6，k3×3	7×7	320	1
9	Conv1×1&Pooling&FC	7×7	1280	1

EfficientNet is based on EfficientNet-B0. The structure of EfficientNet-B0 is shown in Table 1. A total of eight networks from Efficient-Net-B0 to B7 were obtained by model scaling.

In EfficientNet-B0, the network is divided into a total of 9 Stages. Stage 1 is a standard convolutional layer with a convolutional kernel size of 3x3 strides of 2. Stages 2 to 8 repeat the stacked MBConv structures (the Layers in the last column indicate how many MBConv structures there are in that Stage), and Stage 9 has a 1x1 convolutional layer, an average pooling layer, and a fully connected layer composed of them. The first 1x1 convolutional layer in the MBConv expands the channels of the input feature matrix. Each MBConv in the table is followed by a number 1 or 6, representing the multiplicity factor. The k3x3 or k5x5 indicates the size of the convolutional kernel used for the deepwise separable convolution in the MBConv. Resolution indicates the length and width of the feature matrix. The MBConv structure is shown in Fig. 3. MBConv consists mainly of a 1x1 normal convolution (up-dimensional role, containing the BN layer and the Swish activation function), a kxk depth-separable convolution (comprising the BN layer and the Swish activation function), an SE module, a 1x1 standard convolution (down-dimensional role, containing the BN layer), a Dropout layer Composition. The specific *k* values are given in Table I for 3x3 and 5x5 cases.

Deepwise separable convolution was proposed [26] to reduce the number of parameters needed for convolutional computation. With two steps, depthwise and pointwise, more network layers are used to obtain a feature map of the same size dimension as the conventional convolution. EfficientNet uses many depth-separable convolutions in the shallow layers of the network. However, we found that deepwise separable convolution is a "low computation, high access" structure in practice. Most GPU resources are not used for computation but reading and writing data. A regular convolution module is loaded only once. Deepwise separable convolution takes up too much video memory as more modules are loaded more often. Using deepwise separable convolution in the shallow layers of the network can severely impact the training speed. As shown in Fig. 4, the 3×3 convolution in which the depth of the 1×1 convolution kernel used for dimensionality enhancement is detachable was replaced with a normal 3×3 convolution. After replacement, as the number of network layers decreases, the training speed changes significantly.

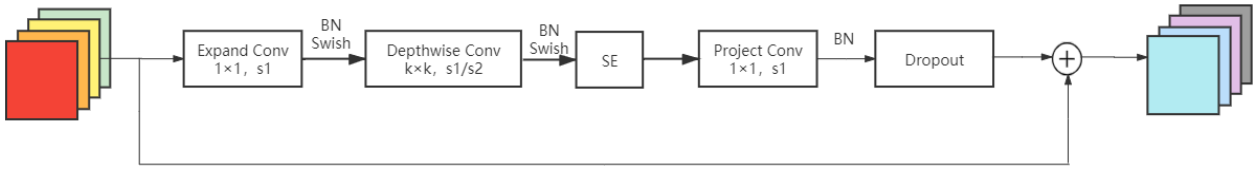

Fig. 3. Structure of MBConv.

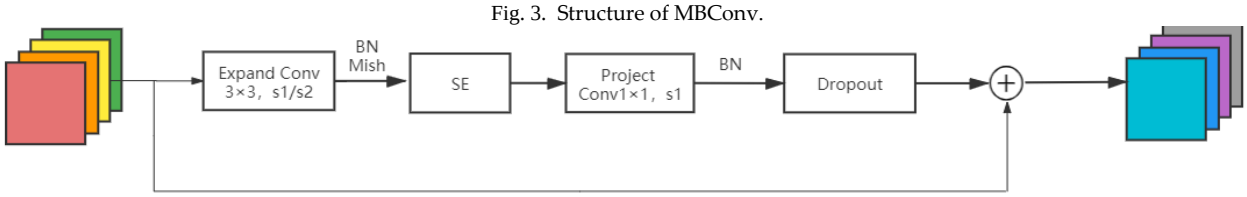

Fig. 4. Improved module structure.

However, suppose all MBConv modules were replaced with this structure. In that case, the entire model will dramatically increase the number of parameters and FLOPs, and the training speed will not change. Therefore, only the MBConv module in stages 2-4 is replaced after experimental testing. The training speed was improved with a small additional cost in terms of parameters and FLOPs. Meanwhile, we used the Mish activation function in the improved EfficientNet. The Mish activation function is effective in many experiments [27] and has been applied as a trick in many detectors, such as YOLOv4 and YOLOv5, which use the Mish activation function in their backbone networks. In the YOLOv4 paper, the Mish activation function is referred to as BoS (Bag of Specials) due to its better performance at a small and almost negligible cost. The formula for the Mish activation function is $f(x) = x \bullet \tanh(\ln(1+e^x))$. The Mish activation function allows better information to penetrate deeper into the neural network when the network layers are deeper. We apply the Mish activation function to EfficientNet to stabilize the network gradient flow and improve accuracy. The improved EfficientNet we call EfficientNet+.

B. Neck Network Skip-BiFPN

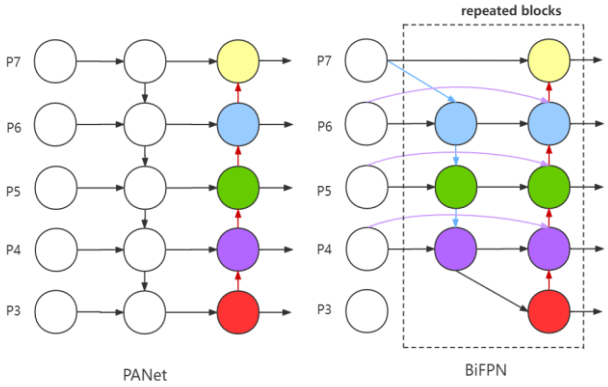

Fig. 5. Feature network design – PANet and BiFPN.

BiFPN is a new feature pyramid network proposed by EfficientDet, which further improves PANet. As shown in Fig. 5, PANet [29] is a bottom-up fusion added on top of the top-down fusion of FPN[30]. On top of PANet, BiFPN removes nodes with only one input edge and introduces cross-node connections at the same level. In addition, the model's is improved by iteratively overlaying the modules without adding additional FLOPs .

Inspired by the idea of same-layer cross-node in BiFPN, cross-layer data flow was added to fully express the semantic information and location information of different layers above and below. The high-resolution feature map was reused, which we believe is vital for detecting small targets. Low-resolution features with complete semantic information were combined with high resolution features with weak semantic details to obtain a feature pyramid with semantic solid information at all scales. EfficientDet believes that each node in BiFPN contributes differently to the whole feature network, so a simple addition cannot be used for feature fusion, and therefore proposes a fast normalization fusion. The weights were assigned to each node according to the weights learned in the network training, and then feature fusion was performed. The fast normalization fusion formulation is $O = \sum_i \frac{\omega_i}{\varepsilon + \sum_j \omega_j} \cdot I_i$. Where the weights $\omega \geqslant 0$ are guaranteed by ReLU. ε = 0.0001, a value to prevent training instability. Taking (1) and (2) as examples, each node has a separate weight corresponding to P_6.

$$P_6^{td} = Conv\left(\frac{\omega_1 \cdot P_6^{in} + \omega_2 \cdot Resize(P_7^{in})}{\omega_1 + \omega_2 + \varepsilon}\right) \quad (1)$$

$$P_6^{out} = Conv\left(\frac{\omega_1' \cdot P_6^{in} + \omega_2' \cdot P_6^{td} + \omega_3' \cdot Resize(P_5^{out})}{\omega_1' + \omega_2' + \omega_3' + \varepsilon}\right) \quad (2)$$

Where P_6^{td} is the intermediate feature at level 6 on the top-down pathway, and P_6^{out} is the output feature at level 6 on the bottom-up pathway. All other features are fused using the same approach. P_6^{td} is summed by multiplying P_6^{in} with the weight ω_1 and then with P_7^{in} multiplied by the weight ω_2 after resize, and finally dividing by the sum of the weights of ω_1 and ω_2. The essence of the fast normalization fusion is that each node has its weight separately so that the information is reasonably distributed through the weights when features are fused. Therefore, we also consider the weight distribution when making cross-layer connections, so we fuse across only one layer of nodes at the outermost nodes. This makes up for the missing information of the original

nodes and ensures that the original feature information weight share of the nodes does not change significantly. At the same time, to make each node input stream more balanced, two data streams, top-down and bottom-up, are introduced in the middle to better compensate for the missing semantic information of the higher-level nodes and the location information of the bottom-level nodes. The distance from the shallowest large scale feature map to the deepest small scale feature map is shortened, and the superficial and high-level feature information is better retained. Using P_6^{out} as an example, the changes in feature fusion after improvement as:

$$P_6^{out} = Conv\left(\frac{\omega_1^{'} \cdot P_6^{in} + \omega_2^{'} \cdot P_6^{td} + \omega_3^{'} \cdot Resize(P_5^{out}) + \omega_4^{'} \cdot Resize[Resize(P_4^{out})]}{\omega_1^{'} + \omega_2^{'} + \omega_3^{'} + \omega_4^{'} + \varepsilon}\right) \quad (3)$$

The improved BiFPN, named Skip-BiFPN and the network structure diagram of Skip-BiFPN is shown in Fig. 6.

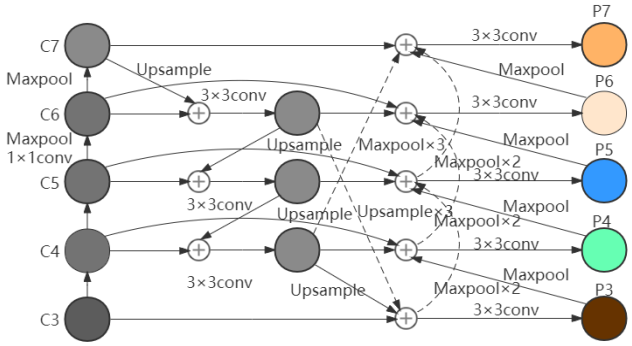

Fig. 6. Structure of Skip-BiFPN.

C. DIoU-NMS

Non-maximal suppression is the most common post-processing method for object detection algorithms. NMS relies on the classifier to obtain multiple detection boxes and confidence that the boxes belong to a category. The confidence levels obtained by the classifier are ranked, the box with the highest score is selected, and all the remaining boxes are iterated through. Finally, the IoU is calculated between all boxes and the current highest scoring box. The boxes are removed if the IoU is greater than a set threshold. This is the traditional NMS algorithm, where the IoU is the only reference indicator and the only suppression principle. If the value of IoU exceeds the threshold we set, then by default, the objects inside both boxes belong to the same category, and then only one box will be left in the end. However, this approach is not well-conveived, and it misses some cases. The formula for IoU is shown in Fig. 7.

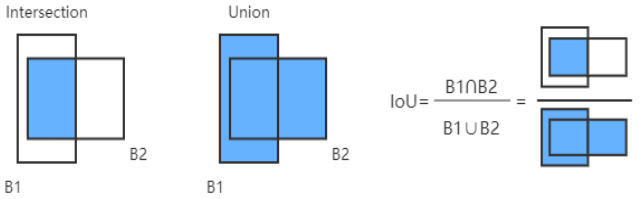

Fig. 7. IoU calculation method.

According to the IoU formula, it can be determined if two objects are close to each other in a real scene, as the IoU between their candidate frames will usually be larger. In this case, they will be considered the same object after NMS processing, and only one detection frame will remain. This can lead to missed detections. In this paper, to effectively avoid such a situation, DioU-NMS was introduced. The traditional IoU calculation of NMS was discarded, and a more comprehensive DioU with improved suppression of candidate frames was used. The DioU was calculated as:

$$R_{DIoU} = \frac{\rho^2(b, b^{gt})}{c^2} \quad (4)$$

Where b and b^{gt} denote the centroids of the two object frames, ρ is the Euclidean distance, and c is the diagonal length of the smallest closed frame covering the two object frames. From the equation, DIoU relies on the overlapping area compared to IoU and considers the distance between the centroids of the two boxes. The distance between the centers of the two object boxes is used to determine whether the two boxes belong to the same object more accurately. The DIoU-NMS equation is in (5).

$$S_i = \begin{cases} S_i, IoU - R_{DIoU}(M, B_i) < \varepsilon \\ 0, IoU - R_{DIoU}(M, B_i) \geq \varepsilon \end{cases} \quad (5)$$

The formula shows that when the overlapping area of two boxes is constant, if the distance between the centroids of the two boxes is more significant and the difference between IoU and DIoU of the highest-scoring prediction box M and the other box B_i is less than a threshold value, the DIoU-NMS will be more inclined to consider that these are two objects and the confidence level S_i of the box B_i will remain the same. Conversely, suppose the distance between the centroids of the two boxes is smaller and the difference between IoU and DIoU of the highest-scoring prediction box M and the other box B_i is greater than a threshold value. In that case, the DIoU-NMS will prefer to think that it is the same object and the S_i value becomes 0, and it is filtered out.

V. Experiments

A. DataSets

The VOC 2012 dataset derived from real scenarios, such as sunny days, cloudy days, nights, and traffic roads, was used. In addition, there are many pedestrians at different scales. Images containing pedestrian Person-like objects were extracted based on the label information, and a total of 8841 images were extracted as the pedestrian detection dataset. In the improved pedestrian detection algorithm model training, the entire dataset was randomly divided into training and test sets according to the ratio of 9:1, for 7961 images in the training set, and 880 images in the test set. The training set was randomly divided into a training dataset and a training validation dataset in the ratio of 9:1.

B. Implementation Details

The proposed pedestrian detection algorithm was experimented on a DELL T7920 graphics workstation based on the PyTorch deep learning framework. The hardware configuration for the experiments was as follows: Intel Xeon E5 series processor, 128G of the server's memory, two

NVIDIA GTX TITAN XP graphics cards with 12G of video memory, Ubuntu 21.04 operating system, Python 3.8, CUDA 10.1, cudnn, PyTorch 1.6.0, torchvision0.7.0, opencv4.4.0, numpy1.18.5, and other Python libraries. A migration study was incorporated into the training process. Freeze training was to speed up the training process and prevent the corruption of the weights in the early training stages. According to the laboratory equipment's hardware conditions and the dataset's size, the training batch size was set to 8, and the epoch to 100. Between 0-50 rounds of freeze training and 51-100 rounds of unfreezing training were used. The learning rate was set to $1e^{-3}$ in the freeze training and $1e^{-4}$ after the thawing. Due to hardware conditions limitations, the Fast-EfficientDet model could only be trained to version D2 in the experiment. If the batch size of Fast-EfficientDet-D3 to D7 were set to 8, it would not train due to insufficient video memory.

C. The Performance Analysis of Fast EfficientDet

This paper introduces the DIoU calculation method in the traditional NMS non-maximum suppression algorithm. In the new suppression method, it is necessary to set a new threshold before the new calculation method can be used to process the frame. This threshold requires constant debugging. As shown in Fig. 8, the threshold hyperparameters were adjusted for multiple DIoUs, to finally obtain the optimal threshold value of 0.45, which provided the best immediate performance. It is worth noting that even if the performance of DIoU-NMS fluctuates slightly during the continuous adjustment of the threshold, its accuracy is always better than the traditional NMS performance before the improvement. Moreover, even if DIoU-NMS does not adjust the threshold, its worst performance is better than the best performance of the original NMS.

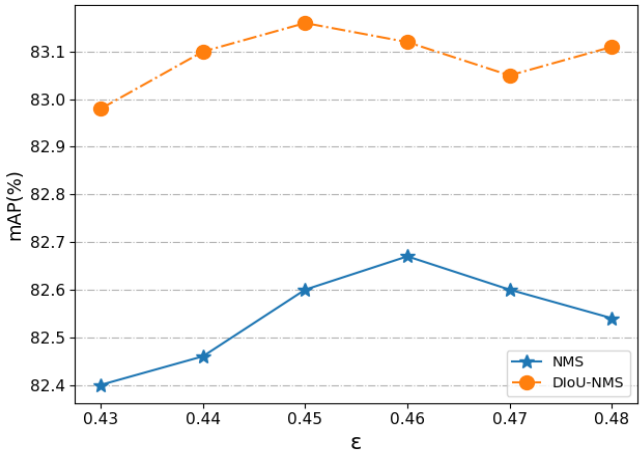

Fig. 8. Comparison of mAP for different thresholds

To improve EfficientNet, improved new modules were used to replace MBConv modules. However, it is impossible to directly decided the number of modules to be replaced in practice. Therefore, experiments were performed to gradually find the best solution for the replacement. Table I shows that the MBConv modules in stages 2-8 were replaced in increasing order. Tests were performed using the EfficientDet-D0 al achieved using the improved modules in stages 2-4. The number of covariates in the backbone network was improved slightly by discarding the deep separable convolution.

TABLE I
THE ACCURACY OF THE IMPROVED MODULE

location	mAP
No Use	82.67%
stage 2-4	82.96%
stage 2-6	82.94%
stage 2-8	81.65%

Under the same conditions, the training speed of the EfficientDet-D0 version with the EfficientNet improvements was tested on the server, as shown in Table II. The improved EfficientDet reveived an algorithm that performed best with a significant increase in training speed due to the introduction of new modules. The training time for 100 epochs was reduced from 865.1 min to 733.3 min, saving 15%. Although the small increase in the number of parameters and computational effort decrease in the inference speed to a certain degree, the model itself meets the requirements of real-time detection, and the significant decrease in training speed reduces the training cost. Concomitantly, the inference speed of the model is limited by the hardware equipment, so the small decrease is negligible.

TABLE II
TRAINING TIME FOR TWO MODELS

Model	Epoch	Batch Size	Train time (min)
EfficientDet-D0	100	8	865.1
Fast EfficientDet-D0	100	8	733.3

To fully demonstrate the performance of the improved model, comparative experiments were performed on three datasets: VOC 2012, Caltech, and INRIA Person. The comparison experiments included various situations, such as a single object, multiple objects, occlusion situations, dense crowds, traffic scenes, and small objects. Figs. 9-11 show the three datasets used for comparison testing, respectively. At the same time, the comparative experiments were enriched to reflect the performance of the improved model more fully. High-resolution pictures of pedestrians in complex scenes were captured by mobile phones, as shown in Fig. 12. The test plots for the original EfficientDet algorithm are shown on the left, and the test plots for the Fast-EfficientDet algorithm are shown on the right. Fig. 9 shows the comparison of the VOC 2012 dataset for small object pedestrians. Since the dataset has a low-resolution image, small objects at a distance can affect the detection accuracy as the constant convolution in the network leads to imperfect object detail information. With Skip-BiFPN, the lack of high-level position information is better compensated. In this case, the improved pedestrian detection algorithm is much better in the face of small object pedestrian detection. In Fig. 10, the Caltech dataset for person-to-person and person-to-object occlusions are compared. With the introduction of DIoU-NMS, the maximum number of pedestrian objects was detected even when they were close together or obscured by objects, effectively reducing the number of missed detections. Fig. 11 compares the INRIA Person dataset for large objects with dense crowd occlusion and pedestrians with different postures. The improved algorithm better detects more pedestrians even with large object crowds obscuring each other. In Fig. 12, athletes on the basketball court and students

Fig. 9. VOC 2012 dataset testing picture.

Fig. 10. Caltech dataset testing picture.

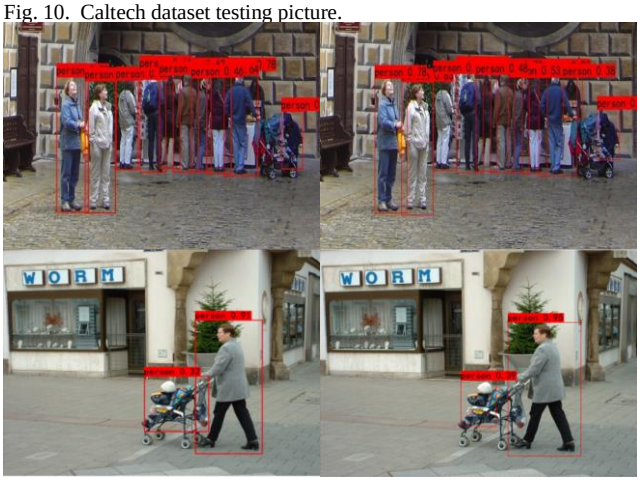

Fig. 11. INRIA Person dataset testing picture.

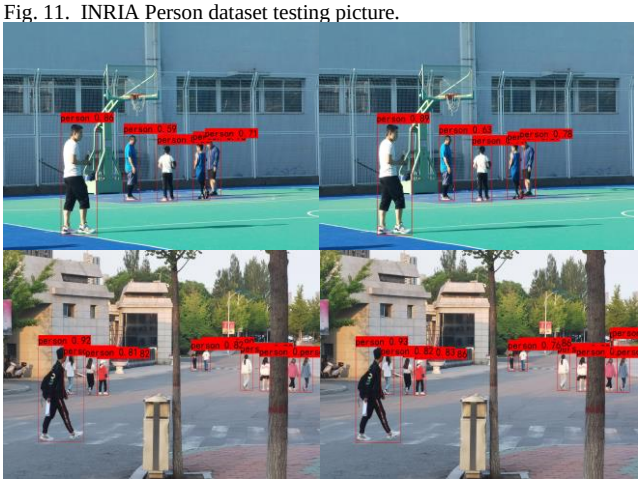

Fig. 12. Complex scenes testing picture.

at the school were filmed by mobile phones. The resolution of the mobile phone images was uniformly set to 1920×1080, much higher than the network input image resolution. After comparative testing, the original EfficientDet algorithm could not detect some occluded people because the target was too small. However, the improved algorithm was able to flag these objects.

D. Comparison of other models

Comparative data tests with other classical algorithms were performed after the comparative experimental analysis with the original algorithm. Fast EfficientDet was compared with the current mainstream object detection algorithms on the same training and test sets, and the results are shown in Table III. AP_{30} is the detection accuracy of the model when the IoU threshold of the detector is 0.3. Similarly, AP_{70} and AP_{70} are the detection accuracies of the model when the IoU threshold is 0.5 and 0.7, respectively. The higher the threshold, the more difficult the network detection. From the table, the Fast EfficientDet algorithm proposed in this paper already outperforms other classical target detection algorithms after version D1. Due to the limited experimental conditions, the experiments can only be done up to version D2. However, it is predicted that the models' performance will improve after D2. Neither the two-stage framework Faster-RCNN, Mask R-CNN [16], the one-stage frameworks YOLOv3, YOLOv4, RetinaNet, TridentNet [16], nor the anchor-free object detector CenterNet [16] can compare with our Fast EfficientDet in terms of accuracy.

E. Ablation Study

A total of four improved modules are proposed in this paper. Ablation experiments were designed to verify the performance improvement of each of the proposed modules on the overall model. The improved modules were added one by one to the original algorithm, and the model was tested with mAP at each stage. The ablation experiments were conducted on EfficientDet-D0. The modules added in the sequence were: EfficientNet+, Skip-BiFPN, DIoU-NMS, and Mish activation function. The experimental results are shown in Table IV. It is clear from the figure that each module improves the performance of the overall model. The BiFPN progress provides the most significant improvement to the overall model. In EfficientDet-D0, the Skip-BiFPN resulted in a 1.31% improvement in the model's mAP. Next, the introduction of DIoU-NMS improved the mAP model by 0.59%. Finally, the EfficientNet+ module and the Mish activation function improved model accuracy by 0.29% and 0.12%.

VI. Conclusion

In this study, an improved EfficientDet pedestrian detection algorithm is proposed. Experimental results show that the accuracy of our model is improved by 2.31%, and the training speed of the model is reduced by 15% compared with the original EfficientDet series algorithm. Moreover, compared to other algorithms, the improved EfficientDet model is more accurate and solves target obscuration better.

However, the model is too deep in network layers, extremely demanding on the hardware, takes up too much video memory during training, and is difficult to train optimally in a typical device environment. Therefore, our

TABLE III
COMPARISON OF STATE-OF-THE-ART MODELS

Model	Backbone	Size	AP_{30}	AP_{50}	AP_{70}
EfficientDet-D0	EfficientNet-B0	512×512	87.40%	82.67%	63.70%
EfficientDet-D1	EfficientNet-B1	640×640	87.62%	84.08%	65.48%
EfficientDet-D2	EfficientNet-B2	768×768	88.64%	85.35%	66.98%
Fast EfficientDet-D0	EfficientNet-B0+	512×512	88.35%	84.98%	67.61%
Fast EfficientDet-D1	EfficientNet-B1+	640×640	89.19%	85.77%	69.62%
Fast EfficientDet-D2	EfficientNet-B2+	768×768	90.35%	86.89%	70.53%
YOLOv3	Darknet-53	416×416	83.66%	81.35%	66.60%
Faster-RCNN	Resnet-50	800×800	84.63%	82.09%	65.15%
RetinaNet	Resnet-101	640×640	86.73%	82.80%	62.02%
CenterNet	Resnet-50	512×512	87.07%	83.12%	62.26%
YOLOv4	CSPDarknet-53	416×416	86.69%	85.09%	68.78%
TridentNet	Resnet-101	640×640	86.12%	84.53%	68.33%
Mask R-CNN	Resnet-101	800×800	84.78%	83.21%	67.26%

TABLE IV
ABLATION STUDY

Backbone EfficientNet	EfficientNet+	Skip-BiFPN	DIoU-NMS	Mish	Epoch	Batchsize	mAP
√					100	8	82.67%
	√				100	8	82.96%
	√	√			100	8	84.27%
	√	√	√		100	8	84.86%
	√	√	√	√	100	8	84.98%

future research direction is to make the network more lightweight without compromising accuracy. Moreover, more pedestrian detection networks will be applied to further improve this issue in the future.

REFERENCES

[1] Vimal S P, Ajay B, "Context pruned histogram of oriented gradients for pedestrian detection," Proceedings of the International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing 2013, 22-23 March 2013, Kottayam, India, pp718-722.

[2] Zhuang J, "Compressive tracking based on HOG and extended Haar-like feature," Proceeding of the IEEE International Conference on Computer and Communications (ICCC) 2016, 14-17 October 2016, Chengdu, China, pp326-331.

[3] Cosma C, Brehar R, Nedevschi S, "Pedestrians detection using a cascade of LBP and HOG classifiers," Proceedings of the IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), 2013, 5-7 September 2013, pp69-75.

[4] Dominguez-Sanchez A, Cazorla M, Orts-Escolano, "Pedestrian Movement Direction Recognition Using Convolutional Neural Networks," IEEE transactions on intelligent transportation systems,2017, vol. 18, no. 12.

[5] Zhao J, Li J, Ma Y, "RPN+ fast boosted tree: Combining deep neural network with traditional classifier for pedestrian detection," Proceedings of the International Conference on Computer and Technology Applications (ICCTA) 2018, 3-5 May 2018, Istanbul, Turkey, pp141-150.

[6] Girshick R, Donahue J, Darrell T, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation," Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2014, pp.580-587.

[7] Ren S, He K, Girshick R, et al."Faster R-CNN:Towards Real-Time Object Detection with Region Proposal Networks,"IEEE transactions on pattern analysis and machine intelligence,2017, vol. 39, no. 6.

[8] Joseph Redmon, Santosh Divvala, Ross Girshick, "You Only Look Once: Unified, Real-Time Object Detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.

[9] Redmon J, Farhadi A, "YOLO9000: Better, faster, stronger," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7263-7271.

[10] Redmon J, Farhadi A. "YOLOv3: An incremental improvement," arXiv:1804.02767, 2018.

[11] Alexey B, Chien-Yao W, Hong-Yuan M L. "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv: 2004.10934, 2020.

[12] Liu W, Anguelov D, Erhan D, et al. "SSD: single shot multibox detector," European Conference on Computer Vision, 2016, pp. 21-37.

[13] Fu C Y, Liu W, Ranga A, et al. "DSSD: Deconvolutional Single Shot Detector," arXiv:1701.06659, 2017.

[14] Li Z, Zhou F. "FSSD: Feature Fusion Single Shot Multibox Detector," arXiv:1712.00960, 2017.

[15] Lin T Y, Goyal P, Girshick R, et al. "Focal Loss for Dense Object Detection," Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988.

[16] Dalal N, Triggs B, "Histograms of Oriented Gradients for Human Detection," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005, pp. 886-893.

[17] Felzenszwalb P F, Girshick R B, McAllester D, et al, "Object detection with discriminatively trained part-based models," IEEE transactions on pattern analysis and machine intelligence, 2010, vol. 32, no. 9.

[18] P Dollár, Tu Z, Perona P, et al. "Integral Channel Features. Proceedings of the British Machine Vision Conference," Proceedings of the British Machine Vision Conference 2009, 7-10 September 2009, London, UK, pp91.1-91.11.

[19] Zhang L, Liang L, Liang X, et al. "Is Faster R-CNN Doing Well for Pedestrian Detection?" Proceedings of the European Conference on Computer Vision 2016, 8-10 October 2016, Amsterdam, Netherlands, pp443-457.

[20] Li J, Liang X, Shen S, et al. "Scale-Aware Fast R-CNN for Pedestrian Detection," IEEE transactions on multimedia, 2018, vol. 20, no. 4.

[21] T. T. Yang, S. Y. Zhou, and A. J. Xu, "Rapid Image Detection of Tree Trunks Using a Convolutional Neural Network and Transfer Learning," IAENG Intl. J. Comput. Sci., vol. 48, no. 2, 2021, pp. 257-265.

[22] P. A. S. Mendes, M. Mendes, A. P. Coimbra, M. M. Crisostomo, "Movement Detection and Moving Object Distinction Based on Optical Flow," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2019, 3-5 July 2019, London, U.K., pp. 48-53.

[23] Yepeng Guan, Wenqing Mao, "Pedestrian Virtual Space Based Abnormal Behavior Detection," IAENG International Journal of Computer Science,2019, vol. 46, no. 2, pp311-320.

[24] Tan M, Pang R, Le Q V, "EfficientDet: Scalable and Efficient Object Detection," Proceeding of the IEEE Conference on Computer Visionand Pattern Recognition (CVPR), 2020, pp.10778-10787.

[25] Tan M, Le Q V, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," Proceeding of the International Conference on Machine Learning 2019, 7-15 June 2019, California, USA, pp6105-6114.

[26] Chollet F, "Xception: Deep Learning with Depthwise Separable Convolutions," Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp.1800-1807.

[27] Misra D. "Mish :A Self Regularized Non-Monotonic Neural Activation Function," arXiv:1908.08681, 2020.

[28] Ramachandran P, Zoph B, Le Q V. “Searching for Activation Functions,” arXiv:1710.05941, 2017.
[29] Liu S, Qi L, Qin H, et al. “Path Aggregation Network for Instance Segmentation,” Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp.8759-8768.
[30] Lin T Y, Dollár, Piotr, Girshick R, et al. “Feature Pyramid Networks for Object Detection,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp.2117-2125.
[31] He K, Gkioxari G, Dollár P, et al. “Mask R-CNN,” Proceeding of the International Conference on Computer Vision, 2017, pp.2980-2988.
[32] Li Y, Chen Y, Wang N, Y Li, et al. “Scale-Aware Trident Networks for Object Detection,” Proceeding of the International Conference on Computer Vision, 2019, pp.6053-6062.
[33] Duan K, Bai S, Xie L, et al. “CenterNet: Keypoint Triplets for Object Detection,” Proceeding of the International Conference on Computer Vision (CVPR), 2019, pp.6568-6577.