# The Study of Performance for Face Detection Based on Multiple Representative Convolutional Neural Networks

Ming Him Foun*

School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland

*Corresponding author: founm@tcd.ie

**Abstract.** Due to the diversity of deep learning models, choosing the suitable model for a specific task can be rather onerous. In this paper, the performance of three deep convolutional neural networks, namely VGG16, ResNet50, and MobileNetV2 on face detection were compared. Each model was trained on a dataset of 11,900 images from the FDDB dataset that included various face sizes and orientations with multiple augmentations, including color alteration, blurring, and flipping. The final layers of the models were modified into a binary classification model and a regression model indicating face found and coordinates of the facial bounding box. The models were trained on the same basis of 40 epochs with batch size 64 with binary cross entropy loss and DIoU loss and a learning rate of 0.0001 with a learning rate decay of 0.8 per epoch. The experimental results demonstrated that VGG16 outperformed ResNet50 and MobileNetV2 in terms of accuracy, with VGG16 achieving the highest $R^2$ score of 0.9240, followed by ResNet50 with a score of 0.8568, and MobileNetV2 with an accuracy of 0.6028. The results suggest that VGG16 is a more suitable choice for face detection applications than ResNet50 and MobileNetV2, while ResNet50 and MobileNetV2 may provide higher accuracy for other image recognition tasks or real time face detections. The findings in this paper can contribute to the selection of appropriate deep learning models for face detection.

**Keywords:** VGG16, ResNet50, MobileNetV2, Face Detection, Deep Learning.

## 1. Introduction

Facial detection is an essential computer vision task which accurately identifies human faces in digital images and video cameras in different angles. Historically, early face detection approaches mainly involve utilizing manual features and heuristics to detect faces, such as the Viola-Jones algorithm. However, these approaches often suffer from poor accuracy and incorrect identifications, especially when detecting faces from different perspectives. During the recent years, this technology has gained significant attention due to its widespread applications in various domains including security, entertainment, and healthcare. Several deep learning approaches, particularly Convolutional Neural Networks (CNNs), have demonstrated promising results in face detection. CNNs are capable of automatically learning advanced features from the raw input data, enabling them to detect faces accurately and robustly from different angles and orientations. One of the major challenges in facial detection is the ability to accurately detect faces from different angles, rotations, tilts, blurriness, and obstructions which can be significant in real-world scenarios where the input data is varied.

Previous research into this field has contributed multiple neural network architectures on image processing and feature extractions [1, 2]. In terms of the previous research, researchers often focus on using one particular architecture, but introducing less on the reasons for selecting such peculiar architecture and comparison analysis of the performances between different neural network architectures. For example, one study focused on using only one architecture, the ResNet-50, for facial detection [3]. This article provided a thorough description on the performance of using ResNet-50 as the base architecture for masked face identification, but it only briefly introduced the efficiency of using the other architectures. While also several studies have investigated face detection using CNNs, most of them focused on detecting faces from frontal views only. However, faces can appear at different angles and orientations during real-world scenarios, resulting a greater challenge to detect them accurately.

In this paper, the focus is about demonstrating the effectiveness of multiple CNN architectures in detecting faces from different perspectives. This research investigates how various CNN architectures perform on the same dataset with multiple augmentations generated for each image. Specifically, this research evaluates the performance of commonly used CNN architectures such as VGGNet, ResNet, and MobileNet on the face detection dataset called FDDB.

To achieve the objective mentioned above, each architecture was trained on the same augmented dataset with the same hyperparameters including the learning rate decay, batch number, and epochs. The approach was evaluated using standard evaluation metrics such as loss function diagrams, test dataset predictions and a comprehensive analysis of the performance between each outcome was introduced. Analysis on the false positive and false negative detections indicated by the wrongly identified faces from the test predictions was conducted and further investigations on the reasons behind them were also performed. Overall, the results of this research contribute to the development of robust face detection systems that can detect faces from different angles and orientations, leading to improved surveillance and facial recognition technologies.

## 2. Method

### 2.1. Dataset description and preprocessing

In this study, the FDDB dataset from the University of Massachusetts was utilized [4]. The original dataset contained a vast collection of 28,204 images at various resolutions, featuring multiple faces captured from different angles, including 2,845 images with ellipse bounding boxes around the identified faces. Fig. 1 provides the example images of the collected dataset. The original labels from the dataset provided the height, width, center X and Y coordinate of the facial ellipse bounding box. These labels have been redesigned into four numbers between 0 to 1, representing the height and width ratio of the X and Y coordinates of the top left corner and the bottom right corner for the rectangle bounding box and stored inside a json file.

Since this study only focuses on images with single face, the dataset was filtered down to only 1700 base total images with labels. First, 70% of the images (1190 images) and their corresponding labels were allocated to the training dataset, while 15% (255 images) with labels were used for the testing and validation datasets, respectively. After the distribution, the Albumentation library was used to augment the base images. A wide variety of augmentation, including vertical flip, horizontal flip, brightness contrast, motion blur, Gaussian blur, Gauss Noise, HSV alteration, and RGB shift, were randomly imposed on each base image. With each base image, 10 augmented images and their corresponding rectangle face bounding box were generated, comprising a sum of 11900, 2550, and 2550 images and labels for the training, testing, and validation dataset. Then, to ensure the size of the images passed into the models were the same, each image was resized to $224 \times 224$ pixels with 3 RGB channels.

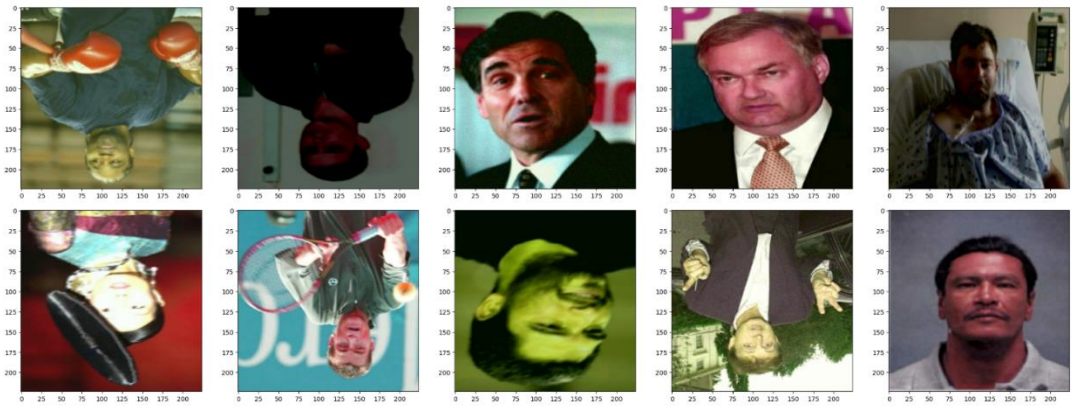

**Fig. 1** Visualizations of Sample Images After Data Augmentation

### 2.2. Proposed Approach

The CNN architectures referred to in this study were the VGG16 model, ResNet50 model, and MobileNetV2 model. VGG16 was introduced in the study which is able to classify 1000 images of 1000 categories with a 92.7% accuracy [5, 6]. VGG16 takes in an input tensor size of 224 × 224 with 3 channels (i.e. RGB) and has 13 convolutional layers, five max pooling layers, and three dense layers. The ResNet50 is a CNN first introduced in the study [7], composing a total of 48 convolutional layers, one max pooling layer, and one average pool layer. In this study, the first 1x1 convolution has one stride and the second 3x3 convolution has two strides for the ResNet50 model [8], differing to the backbone structure mentioned in the paper. MobileNetV2 is a CNN network composed of the initial fully convolution layer with 32 filters along with 19 residual bottleneck layers, referenced in the study [9].

To satisfy the training requirement based on the current dataset, the top layers of the models were being modified into linear layers with the final output dimensions 1 and 4, depicting a binary classification model to determine whether a face was found and a regression model to accurately draw the bounding box of the face with the X and Y coordinates of the top left corner and the bottom right corner.

### 2.3. Implementation Details

The binary cross entropy and DIoU loss function was used for the binary classification and the regression model, respectively [10]. The binary cross entropy loss function evaluates the difference between the predicted probability distribution and the true probability distribution of the binary output, computing the cross-entropy loss between the true labels and the predicted labels. The DIoU loss function is an improvement over the IoU loss function which takes into account the distance between the predicted bounding box and the ground truth bounding box, calculated by the ratio of the square of the Euclidean distance between the center points over the square of the smallest enclosing box that contains both boxes.

$$\text{Binary cross entropy loss} = -(y \cdot log(y) + (1-y) \cdot log(1-y)) \tag{1}$$

$$intersection = max(0, (min(x_2, x_2) - max(x_1, x_1)) \cdot (min(y_2, y_2) - max(y_1, y_1))) \tag{2}$$

$$union = (x_2 - x_1) \cdot (y_2 - y_1) + (x_2 - x_1) \cdot (y_2 - y_1) - intersection \tag{3}$$

$$center = \left[\left(x_1 + \frac{(x_2 - x_1)}{2}\right) - \left(x_1 + \frac{(x_2 - x_1)}{2}\right)\right]^2 + \left[\left(y_1 + \frac{(y_2 - y_1)}{2}\right) - \left(y_1 + \frac{(y_2 - y_1)}{2}\right)\right]^2 \tag{4}$$

$$box = (max(x_2, x_2) - min(x_1, x_1))^2 + (max(y_2, y_2) - min(y_1, y_1))^2 \tag{5}$$

$$DIoU\ loss = 1 - \frac{intersection}{union} + \frac{center}{box} \tag{6}$$

The total loss was calculated as the sum of half the classification loss and half the regression loss. The Adam optimizer was used with a learning rate of 0.0001 and a step learning rate decay of 0.8 after each epoch. Each model was trained with 40 epsochs with batch size 64 and every epoch being tuned on the validation dataset to improve the model's accuracy on the new data. The training accuracy and the testing accuracy were evaluated using the $R^2$ score for each model.

## 3. Results and Discussion

**Table 1**. The performance of the models in terms of the accuracy

| Model | Training Accuracy | Testing Accuracy |
|---|---|---|
| VGG16 | 0.9264 | 0.9240 |
| ResNet50 | 0.8398 | 0.8568 |
| MobileNetV2 | 0.7091 | 0.6028 |

By computing the $R^2$ score on the predicted bounding box, each model was experimented on the training dataset with 40 epochs of batch size 64. The result demonstrated that the VGG16 can achieve the best performance with an accuracy of 0.9240 on the test dataset shown in Table 1, followed by ResNet50 with a testing accuracy of 0.8568 and eventually the MobileNetV2 model with an accuracy of 0.6028.

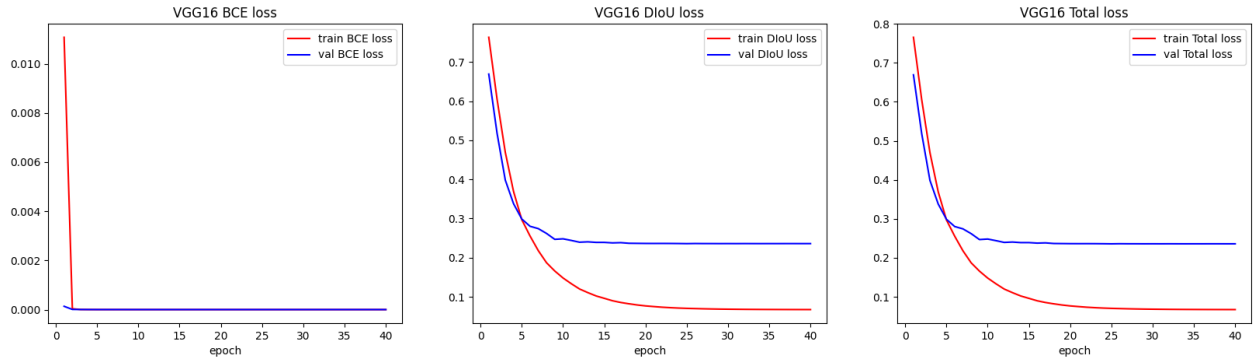


**Fig. 2** Training Loss Curve of VGG16

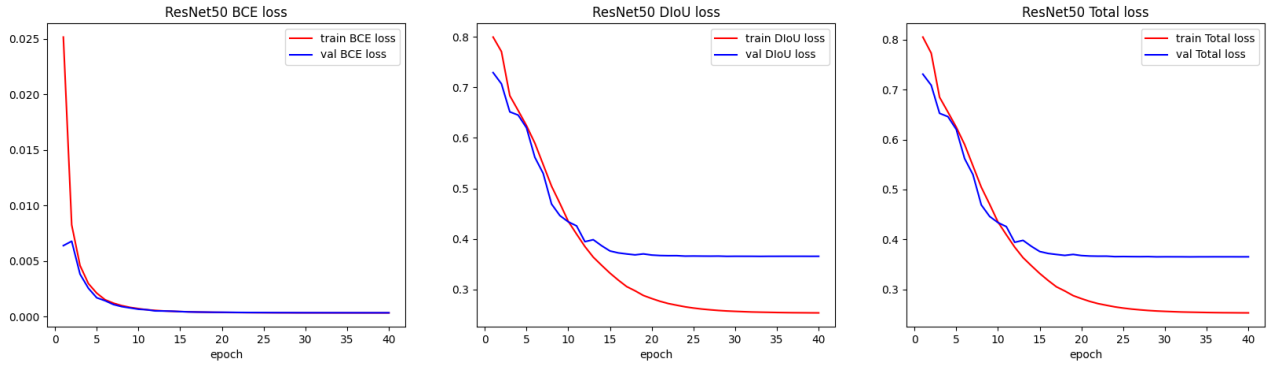


**Fig. 3** Training Loss Curve of ResNet50

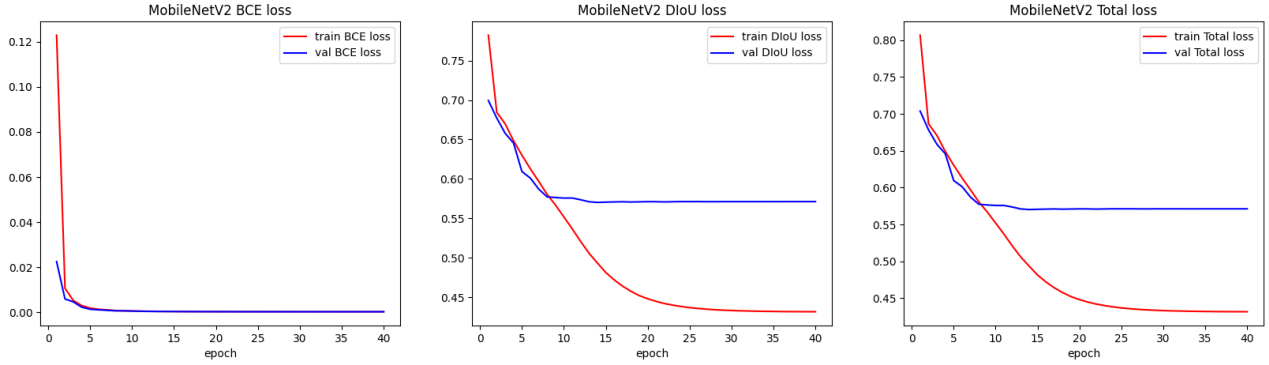


**Fig. 4** Training Loss Curve of MobileNetV2

As it was shown in Fig. 2, Fig. 3 and Fig. 4, the binary cross entropy loss for every model has dropped to a minimum, approaching close to 0. However, the regression loss varied by dropping to different values after being trained with 40 epochs of batch size 64, with VGG16 being 0.067, ResNet50 being 0.2533, and MobileNetV2 being 0.4317.

The figures displayed a smooth curve during the training process without peaks, indicating a smooth training progress without major problems with the dataset, model architecture, or parameters. With the regression loss being varied and the classification loss dropping drastically to approaching 0, the total loss curve indicated a similar image to the regression loss curve. Viewing the loss from the figures and the accuracy, VGG16 demonstrated the best performance in finding the bounding box for faces with the highest accuracy with the lowest loss, followed by ResNet50 and the last, MobileNetV2. VGG16 has shown a fast convergence in its regression loss towards the optimal solution, attaining values under 0.1 after 15 epochs. ResNet50 coverages under the value 0.3 at 18 epochs, but loss reduction gradually decreases for the remaining epochs ending up with a regression loss of 0.2533. MobileNet50 converges at 0.45 at epoch 20, showing very limited reduction in regression loss with a very slow training process for the subsequent epochs, causing the model to underfit and resulting a poor performance on the accuracy of finding the facial bounding boxes.

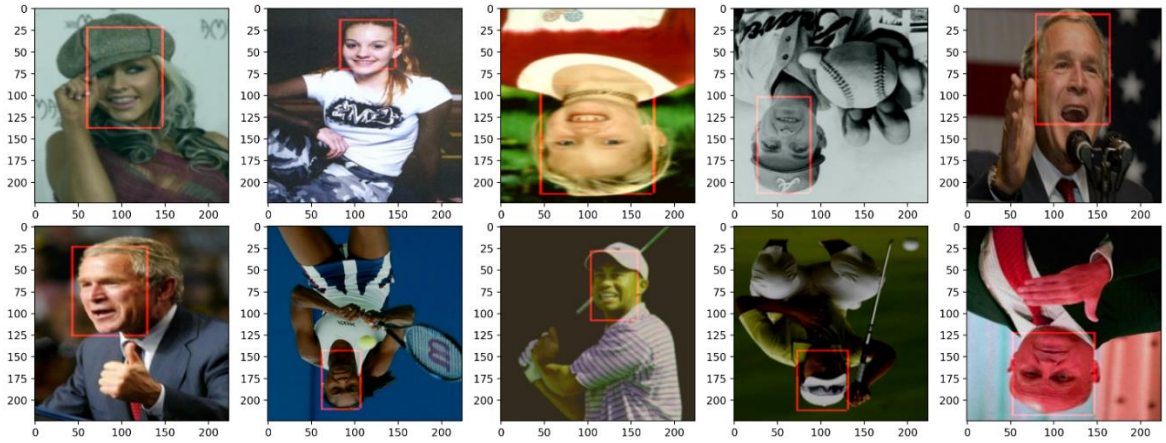

**Fig. 5** Prediction bounding box images of VGG16

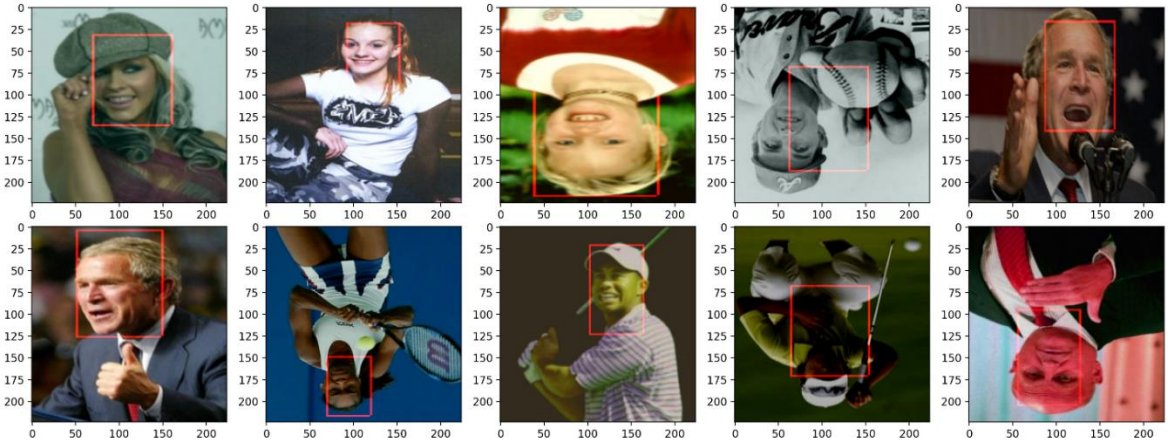

**Fig. 6** Prediction bounding box images of ResNet50

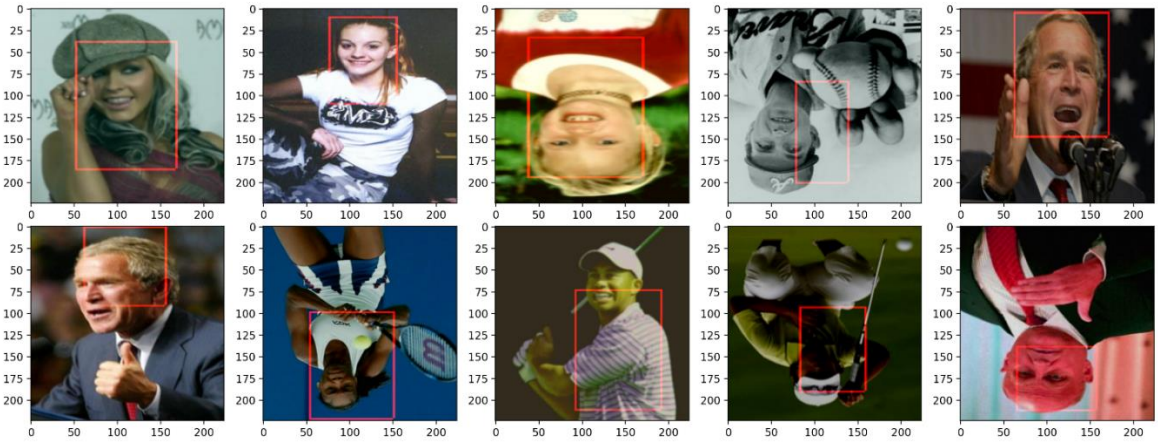

**Fig. 7** Prediction bounding box images of MobileNetV2

The trained models were used for predictions on the test dataset to find the facial bounding boxes. Taking 10 samples from the test dataset and drawing the boxes for each model, the difference was significantly displayed. VGG16 with a notable accuracy of 0.9240 found the correct bounding boxes for every face displayed in Fig. 5 when the pictures were augmented. ResNet50, displayed in Fig. 6, showed some difficulty in accurately identifying faces when images were augmented, illustrating the incorrect bounding boxes for the two images on the fourth column and additionally the last image, but properly determined the bounding box for other faces, ensuing a testing accuracy of 0.8568. Lastly, in Fig. 7, MobileNetV2 to some extent labelled the bounding box including the face, but the accuracy to draw boxes closely around the face was insignificant, resulting in a testing accuracy of 0.6028 which is unfavorable. This experiment has demonstrated that the CNN architecture VGG16 is the most ideal architecture in face detection compared to ResNet50 and MobileNetV2.

## 4. Conclusion

In this study, the accuracy performance comparison of three prevalent CNN architectures including VGG16, ResNet50, MobileNetV2 on face detection was carried out. Each model was modified to output a binary classification and a regression to find the face and the corresponding bounding box. Loss function graphs and prediction results on the test dataset were implemented to evaluate the proposed approach. The outcomes achieved by computing the $R^2$ score on the testing dataset, with VGG16 acquired the highest accuracy of 0.9240, followed by ResNet50 with a score of 0.8568, and ultimately MobileNetV2 with a score of 0.6028. Thus, VGG16 has proved to be an appropriate option for facial detection tasks rather than ResNet50 or MobileNetV2. This study highlights the importance of selecting the appropriate deep learning model for a specific task whereas ResNet50 and MobileNetV2 may perform better on other image recognition tasks. Future research endeavors will aim to compare the performance of additional CNN architectures on larger and more diverse datasets with multiple faces and more profound augmentations.

## References

[1] Zhu Y, Cai H, Zhang S, et al. Tinaface: Strong but simple baseline for face detection. arXiv preprint arXiv:2011.13183, 2020.

[2] Chi C, Zhang S, Xing J, et al. Selective refinement network for high performance face detection, Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 8231-8238.

[3] Mandal B, Okeukwu A, Theis Y. Masked face recognition using resnet-50. arXiv preprint arXiv:2104.08997, 2021.

[4] FDDB: A Benchmark for Face Detection in Unconstrained Settings. Technical Report UM-CS-2010-009, Dept. of Computer Science, University of Massachusetts, Amherst. 2010.

[5] Simonyan K, & Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[6] "ImageNet." [Online]. Available: http://image-net.org/index. [Accessed: 20-Febuary-2023].

[7] He K, Zhang X, Ren S, & Sun J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778), 2016

[8] https://pytorch.org/hub/nvidia_deeplearningexamples_resnet50/ [Accessed: 18-March-2023]

[9] Sandler M, Howard A, Zhu M, Zhmoginov A, & Chen L C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520), 2018.

[10] Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: Faster and better learning for bounding box regression, Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 12993-13000.