IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 38, No. 4, August, 2021, pp. 1253-1257
Journal homepage: http://iieta.org/journals/ts

Integration Between Cascade Region-Based Convolutional Neural Network and Bi-Directional Feature Pyramid Network for Live Object Tracking and Detection

Lehai Zhong[1], Jiao Li[1*], Feifan Zhou[3], Xiaoan Bao[2], Weiyin Xing[1], Zhengyong Han[1], Jinsheng Luo[1]

[1] School of Electronics and Information, Mianyang Polytechnic, Mianyang 621000, China
[2] School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China
[3] School of Electrical Engineering, Southwest Jiaotong University, Chengdu 610031, China

Corresponding Author Email: lijiao@mypt.edu.cn

https://doi.org/10.18280/ts.380437

Received: 5 March 2021
Accepted: 8 June 2021

Keywords:
cascade region-based convolutional neural network (R-CNN), bi-directional feature pyramid network (BiFPN), live object tracking and detection

ABSTRACT

The current target tracking and detection algorithms often have mistakes and omissions when the target is occluded or small. To overcome the defects, this paper integrates bi-directional feature pyramid network (BiFPN) into cascade region-based convolutional neural network (R-CNN) for live object tracking and detection. Specifically, the BiFPN structure was utilized to connect between scales and fuse weighted features more efficiently, thereby enhancing the network's feature extraction ability, and improving the detection effect on occluded and small targets. The proposed method, i.e., Cascade R-CNN fused with BiFPN, was compared with target detection algorithms like Cascade R-CNN and single shot detection (SSD) on a video frame dataset of wild animals. Our method achieved a mean average precision (mAP) of 91%, higher than that of SSD and Cascade R-CNN. Besides, it only took 0.42s for our method to detect each image, i.e., the real-time detection was realized. Experimental results prove that the proposed live object tracking and detection model, i.e., Cascade R-CNN fused with BiFPN, can adapt well to the complex detection environment, and achieve an excellent detection effect.

1. INTRODUCTION

In recent years, significant progress has been made in the detection of live objects like human and other animals, which provides technical support to semantic segmentation, cross-border tracking, and behavioral analysis. The relevant detection technologies have been widely applied in multiple fields, such as public security, image retrieval, as well as animal tracking and protection. However, the detection accuracy might be dragged down, if the target is occluded or small. To improve detection accuracy, target detection researchers are striving to mitigate the mistakes and omissions of live object detection [1].

Traditional target detection methods mainly rely on machine learning. The detection algorithms can be decomposed into three phases: region selection, feature extraction, and target classification. During region selection, multiple candidate regions are obtained through sliding window operation or selective search. During feature extraction, the features of each candidate region are extracted through scale-invariant feature transform (SIFT) [2], and histogram of oriented gradients (HOG) [3]. During target classification, the targets are classified according to the features extracted in the second phase. The main classifiers are to support vector machine (SVM) [4], and adaptive boosting (AdaBoost) [5]. For live object detection, the quality of feature extraction directly determines the detection accuracy of small or occluded targets. The traditional target detection methods are inaccurate, and time-consuming, with a high omission rate. Considering the actual needs of live object detection, this paper tries to improve the phase of feature extraction.

The further development of deep learning (DL) enables the continuous improvement of the accuracy and speed of target detection algorithms, with the aid of multi-layer computing networks. More and more target detection methods have emerged based on deep convolutional neural networks (DCNNs) [6-9]. There are mainly two types of DL-based target detection networks: one-stage detection algorithms, and two-stage detection algorithms. One-stage detection algorithms directly generate the class probability and coordinates of the object, without needing region selection. Typical examples include single shot detection (SSD) and its derivatives [10], you only look once (YOLO) series [11], etc. Two-stage detection algorithms first perform region selection, and then classify candidates regions. Typical examples include region-based CNN (R-CNN) series [12], and various improved versions of R-CNN, namely, Fast R-CNN [13], and Faster R-CNN [14]. Cascade R-CNN [15] is a multistage extension of Faster R-CNN. Two-stage algorithms have lower error rate and omission rate than one-stage algorithms. Nevertheless, the above algorithms do not make full use of the context of the region of interest (ROI), and ignore the importance of shallow position information to the detection of small or occluded targets.

Taking Cascade R-CNN as the basic algorithm, this paper integrates bi-directional feature pyramid network (BiFPN) to enhance the bidirectional feature fusion. The integrated network fully utilizes shallow position information, which facilitates the detection of small or occluded targets, to improve the detection accuracy of occluded or small targets, and to reduce detection mistakes and omissions [16]. The proposed algorithm was compared with original Cascade R-

CNN algorithm on a dataset of complex video frames taken in a natural reserve. The results show that our algorithm achieved a much higher mean average precision (mAP) than the original Cascade R-CNN.

2. CASCADE R-CNN TARGET DETECTION ALGORITHM

Cascade R-CNN cascades three detection networks to constantly optimize the prediction result. Unlike ordinary cascade, the three detection networks of Cascade R-CNN are obtained through training on positive and negative samples, which is determined based on different intersection over union (IoU) thresholds. For different detectors and different benchmark networks, the cascade process steadily improves the average precision (AP). The improvement increases with the IoU threshold.

The target detection model based on Cascade R-CNN is illustrated in Figure 1, where H1-3 represents the detection and classification header of Faster R-CNN, B1-3 represents the detection boxes after regression. The network operation can be summarized as: B1 regression outputs a detection box, which is imported to H2; Then, B2 regression outputs another box, which is imported to H3; Finally, C1-3 outputs the specific classes of the samples [15]. The three cascaded classifiers, whose threshold increase progressively, steadily improves the positive sample quality of each detector through repeated classifications and regressions, thereby enhancing detection accuracy continuously.

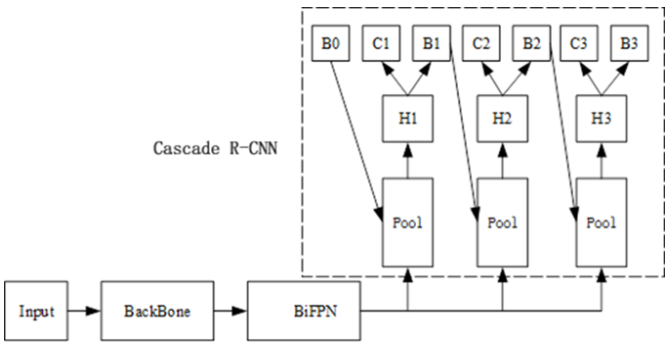


Figure 1. Structure of target detection model based on Cascade R-CNN

3. BIFPN MULTI-FEATURE FUSION

The classifiers trained on a single feature cannot fully utilize the context of ROI, and ignores the position information to the detection of small or occluded targets, bringing a low detection accuracy and a high omission rate. What is worse, the background noise in the complex environment suppresses the quality of feature data, adding to the difficulty in classifier training and reducing the classification accuracy. The main solution to these problems is to implement feature fusion: extracting various features for classifier training, and make up for the inherent defect of a single feature through the complementarity between features.

In 2017, Lin et al. [17] proposed a novel detection model called FPN, which greatly improves the target detection accuracy by combining Faster R-CNN with up-sampling structure, and fusing the feature information of high- and low-level feature maps. FPN is now the most widely used multiscale fusion method. Recently, a series of cross-scale feature fusion methods have been developed, including path aggregation network (PANet) [18] and neural architecture search (NAS)-FPN [19].

The earliest approach of feature fusion simply adds up the features. However, different features have different resolutions, and make different contributions to the output feature. To solve this problem, Tan et al. [20] put forward the BiFPN in 2020, in which learnable weight is used to evaluate the importance of each feature and learn different features, and feature fusion is implemented repeatedly from top to bottom and from bottom to top.

Following the idea of two-way fusion, BiFPN constructs a top-down channel, and a bottom-up channel. For the multiscale information from the backbone network, the resolution scales of the features are unified through up-sampling and down-sampling. Besides, horizontal connections are added between features of the same scale to ease the feature information loss induced by the excessive number of network layers. The BiFPN structure is explained in Figure 2.

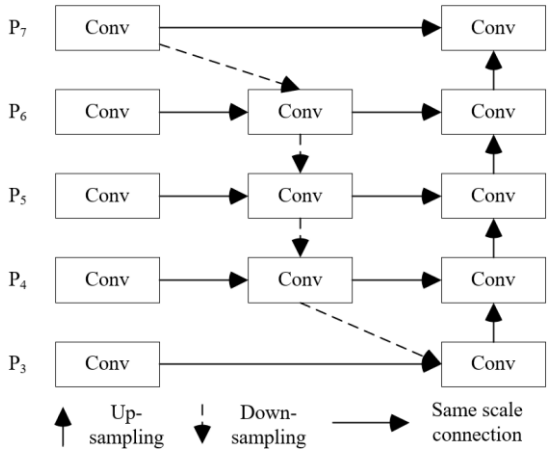


Figure 2. BiFPN structure

4. MODEL IMPROVEMENT

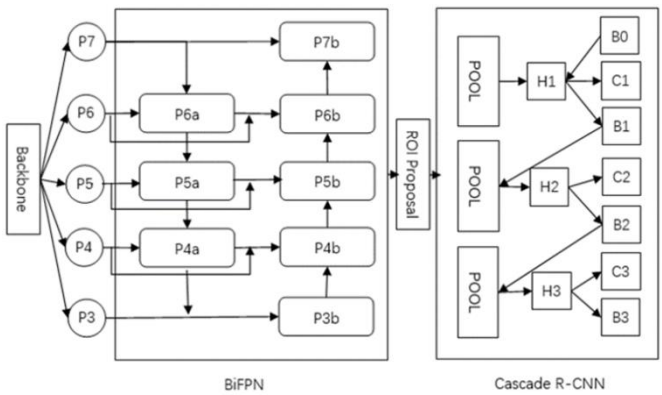


Figure 3. Algorithm's structure

BiFPN fully fuses various features, especially those of occluded or small live objects in a complex background. The top-down feature fusion is combined with bottom-up feature fusion to embed bottom-level environmental features in high-level features. In this way, the loss of shallow features is prevented, and the foreground is effectively differentiated

from the background. As a result, BiFPN can detect small or occluded targets at a high accuracy, and achieve a good result by integrating segmentation task with detection task.

Figure 3 shows the structure of our target detection module, which integrates BiFPN into Cascade R-CNN. In the backbone phase, five strides (8, 16, 32, 64, and 128) are selected to constitute an FPN. The resulting features of different resolutions are imported into BiFPN model as inputs P3, P4, P5, P6, and P7.

The BiFPN model first up-samples P7, and stacks the result with P6 to obtain P6a; next, the model up-samples P6a, and stacks the result with P5 to obtain P5a; after that, the model up-samples P5a, and stacks the result with P4 to obtain P4a; then, the model up-samples P4a, and stacks the result with P3 to obtain P3b. Hence, P3b realizes the cross-scale connection and weighted feature fusion between P3 and P4a.

After P3b, P4a, P5a, and P6a are obtained, P3b is down-sampled, and the result is stacked with P4a and P4 to obtain P4b. Thus, P4b realizes the cross-scale connection and weighted feature fusion between P3b, P4, and P4a. Next, P4b is down-sampled, and the result is stacked with P5a and P5 to obtain P5b. Then, P5b realizes the cross-scale connection and weighted feature fusion between P4b, P5 and P5a. After that, P5b is down-sampled, and the result is stacked with P6 and P6a to obtain P6b. Then, P6b realizes the cross-scale connection and weighted feature fusion between P5b, P6 and P6a. Thereafter, P6b is down-sampled, and the result is stacked with P7 to obtain P7b. Then, P7b realizes the cross-scale connection and weighted feature fusion between P6b, P7 and P7a.

The feature maps on the layers of P3-7 are subjected to cross-scale connection and weighted feature fusion in BiFPN. In this way, the multiscale features are extracted from the input image. BiFPN implements rapid normalized fusion:

$$O = \sum_i \frac{w_i}{\varepsilon + \sum_j w_j} \cdot I_i \tag{1}$$

where, ε =0.0001; w_i is a nonnegative learnable weight. Rectified linear unit (ReLU) is adopted to ensure the data stability. Each normalized weight falls between 0 and 1.

Taking the sixth layer of BiFPN as an example, the intermediate layer P_6^{td} in the top-down direction of P_6 outputs the following feature:

$$P_6^{td} = Conv\left(\frac{w_1 \cdot P_6^{in} + w_2 \cdot Resize(P_7^{in})}{w_1 + w_2 + \varepsilon}\right) \tag{2}$$

The output layer P_6^{out} in the bottom-up direction of P_6 outputs the following feature:

$$P_6^{out} = Conv\left(\frac{w_1' \cdot P_6^{in} + w_2' \cdot P_6^{td} + w_3' \cdot Resize(P_5^{out})}{w_1' + w_2' + w_3' + \varepsilon}\right) \tag{3}$$

The other layers of BiFPN are constructed in a similar manner.

When it comes to the ROI proposal phase, multiple feature maps of different sizes are screened on different layers, and imported into Cascade R-CNN. In Cascade R-CNN model, the least absolute deviation (LAD) loss function is the same as Fast R-CNN:

$$R_{loc}[f] = \sum_{i=1}^{N} L_{loc}(f(x_i, b_i), g_i) \tag{4}$$

Through continuous iteration, B2 is initialized with B1, and B3 is initialized with B2:

$$f'(x, b) = f \circ f \circ \cdots \circ f(x, b) \tag{5}$$

Setting the initial threshold to 0.5, detectors with a progressively increasing threshold are adopted for repeated classifications and regressions, such that the positive sample quality steadily improves for each detector.

The cross-entropy loss is chosen as the classification loss function:

$$R_{cls}[h] = \sum_{i=1}^{N} L_{cls}(h(x_i), y_i) \tag{6}$$

The loss function after cascading can be expressed as:

$$\begin{aligned} L(x^t, g) &= L_{cls}(h_t(x^t), y^t) \\ &+ \lambda[y^t \geq 1] L_{cls}(f_t(x^t, b^t), g), b^t = f_{t-1}(x^{t-1}, b^{t-1}) \end{aligned} \tag{7}$$

5. EXPERIMENTS AND RESULTS ANALYSIS

5.1 Experimental environment and parameter setting

Our experiments were carried out with NVIDIA Tesla V100 SXM2 graphics processing unit (GPU), Ubuntu Linux Server 16.04, and PyTorch 1.2 with CUDA Toolkit 10.1. The training set is in VOC format. The batch size was set to 8, momentum to 0.9, non-maximum suppression to 0.5, and maximum epoch to 24. The learning rate was configured by the on-demand adjustment strategy. During the experiments, the learning rate was initialized as 0.005, and attenuated in the 16th and 22nd epochs with an attenuation coefficient of 0.001. The network parameters were optimized through stochastic gradient descent, with a weight attention coefficient of 0.0001. The model detection effect was evaluated by mAP.

5.2 Experimental process

To evaluate the performance of our algorithm in complex environments, this paper prepares a dataset with the tracking and monitoring video frames of wild animals. A total of 4,000 images on 10 types of animals were selected, and randomly divided into a training set (3,200) and a test set (800) at the ratio of 4: 1.

The individual animals were labeled in the sample images with the open annotation tool LabelMe, including occluded animals, small targets like birds, and nocturnal animals. The brightness of the sample images was randomly adjusted to create more samples with different light conditions. Besides, the sample images were rotated by different degrees to simulate the shooting effect of cameras from different angles. Further, the sample images were mosaiced randomly to emulate the occlusion of animals.

After that, a target detection model was established by integrating BiFPN structure into Cascade R-CNN. Then, the labeled training set was imported to Cascade R-CNN. The convergence of each training was judged against the training loss. The calculation was performed iteratively until the loss converged, producing the final model.

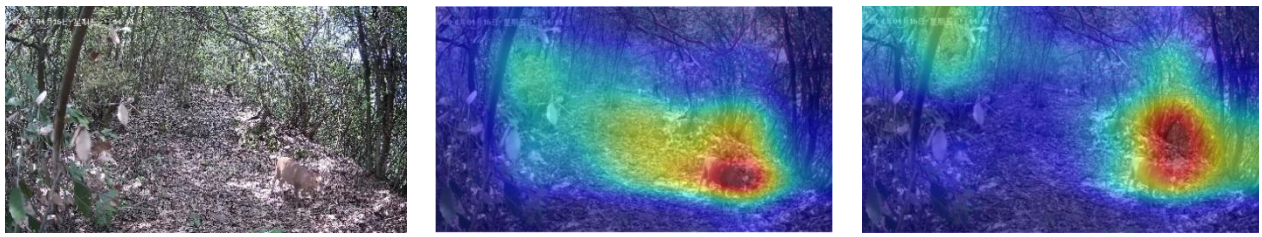
5.3 Results analysis

SSD was chosen as the representative of traditional target detection algorithm. Both Cascade R-CNN and SSD rely on

the backbone network of VGG-16. Then, the detection results of SSD, Cascade R-CNN, SSD coupled with BiFPN, and Cascade R-CNN coupled with BiFPN (our algorithm) are compared in Table 1. It can be inferred that the network models centering on Cascade R-CNN were superior in mAP than SSD-centered models. The mAP of the original Cascade R-CNN was 86.4%, while that of BiFPN Cascade R-CNN rose to 91%; it only took 0.42s for BiFPN Cascade R-CNN to detect an image. The comparison demonstrates the good recognition effect, detection accuracy, and real-time performance of BiFPN Cascade R-CNN for live object tracking and detection. Figure 4 compares the detection results of Cascade R-CNN and BiFPN Cascade R-CNN on animal images. Figures 5-7 present the recognition results on small animals, partly occluded animals, and nocturnal animals, respectively.

Table 1. Detection results of SSD, Cascade R-CNN, SSD coupled with BiFN, and Cascade R-CNN coupled with BiFPN

Model	Backbone network	mAP	Detection time
SSD	VGG-16	82.6%	0.3s
BiFPN SSD	VGG-16	83.8%	0.41s
Cascade R-CNN	VGG-16	84.6%	0.36s
BiFPN Cascade R-CNN	VGG-16	91%	0.42s



(a) Original image (b) Result of Cascade R-CNN (c) Result of BiFPN Cascade R-CNN

Figure 4. Detection results of Cascade R-CNN and BiFPN Cascade R-CNN on animal images

Figure 5. Recognition results on small animals

Figure 6. Recognition results on partly occluded animals

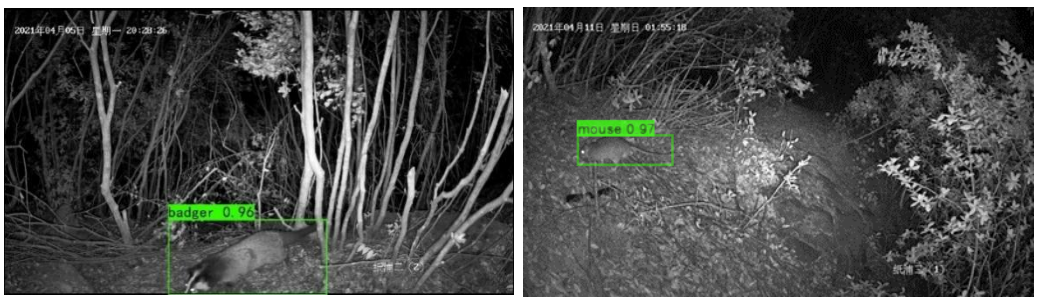


Figure 7. Recognition results on nocturnal animals

6. CONCLUSIONS

The current target tracking and detection algorithms often have mistakes and omissions when the target is occluded or small. To solve the problems, this paper integrates BiFPN into Cascade R-CNN for live object tracking and detection. The addition of BiFPN strengthens the fusion of two-way features, and makes full use of shallow position information, which benefits the detection of small or occluded targets. In this way, the detection accuracy is improved for occluded and small targets. Specifically, the mean precision and speed of detection and classification are improved, and the problems of classifiers trained on a single feature are overcome (low precision, high false positives, and high omission rate). Finally, the proposed live object tracking and detection model, i.e., BiFPN Cascade R-CNN, was applied to detect wild animals, and compared qualitatively with SSD, Cascade R-CNN, etc. The experimental results show that the proposed live object tracking and detection model, i.e., Cascade R-CNN fused with BiFPN, can adapt well to the complex detection environment, realize a higher mAP (91%) than other algorithms, and satisfy the demand for real-time detection.

ACKNOWLEDGEMENTS

This paper is supported by Key Research and Development Project, Science and Technology Plan, Sichuan Province, China (Grant No.: 2019YFG0112); Key Industrial Program of Major Special Science and Technology Project, Science and Technology Plan, Zhejiang Province, China (Grant No.: 2014C01047).

REFERENCES

[1] Liu, B.W., Peng, Z.L., Fan, C.A. (2020). Pedestrian detection method based on Cascade-Rcnn. Wuxian Hulian Keji, 17(2): 15-17. http://dx.chinadoi.cn/10.3969/j.issn.1672-6944.2020.02.007

[2] Lowe, D.G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2): 91-110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[3] Dalal, N., Triggs, B. (2005). Histograms of oriented gradients for human detection. 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), San Diego, CA, USA, pp. 886-893. https://doi.org/10.1109/CVPR.2005.177

[4] Cortes, C., Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3): 273-297. https://doi.org/10.1007/BF00994018

[5] Freund, Y., Schapire, R.E. (1996). Experiments with a new boosting algorithm. Machine Learning: Proceedings of the Thirteenth International Conference, pp. 148-156.

[6] Huang, J., Zhang, G. (2020) Survey of object detection algorithms for deep convolutional neural networks. Computer Engineering and Applications, 56(17): 12-23. http://dx.chinadoi.cn/10.3778/j.issn.1002-8331.2005-0021

[7] Luo, H., Peng, S., Chen, H.K. (2021) Review on latest research progress of challenging problems in object detection. Computer Engineering and Applications, 57(5): 36-46. http://dx.chinadoi.cn/10.3778/j.issn.1002-8331.2011-0205

[8] Fang, L.P., He, H.J., Zhou, G.M. (2018). Research overview of object detection methods. Computer Engineering and Applications, 54(13): 11-18, 33. http://dx.chinadoi.cn/10.3778/j.issn.1002-8331.1804-0167

[9] Li, Z.W., Hu, A.S., Wang, X.F. (2020). Survey of vision based object detection methods. Computer Engineering and Applications, 56(8): 1-9. http://dx.chinadoi.cn/10.3778/j.issn.1002-8331.2001-0163

[10] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C. (2016). SSD: Single shot multibox detector. European Conference on Computer Vision, Amsterdam, The Netherlands, pp. 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[11] Redmon, J., Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[12] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, pp. 580-587. https://doi.org/10.1109/CVPR.2014.81

[13] Girshick, R. (2015). Fast r-CNN. Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, pp. 1440-1448. https://doi.org/10.1109/ICCV.2015.169

[14] Ren, S., He, K., Girshick, R., Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6): 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

[15] Cai, Z., Vasconcelos, N. (2018). Cascade R-CNN: Delving into high quality object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 6154-6162. https://doi.org/10.1109/CVPR.2018.00644

[16] Li, S.J., Wu, N., Wang, P., Li, H.L. (2021). Vehicle target detection method based on improved cascade RCNN. Computer Engineering and Applications, 57(5): 123-130. http://dx.chinadoi.cn/10.3778/j.issn.1002-8331.2005-0416

[17] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S. (2017). Feature pyramid networks for object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, USA, pp. 2117-2125. https://doi.org/10.1109/CVPR.2017.106

[18] Liu, S., Qi, L., Qin, H., Shi, J., Jia, J. (2018). Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 8759-8768. https://doi.org/10.1109/CVPR.2018.00913

[19] Ghiasi, G., Lin, T.Y., Le, Q.V. (2019). NAS-FPN: Learning scalable feature pyramid architecture for object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, pp. 7036-7045. https://doi.org/10.1109/CVPR.2019.00720

[20] Tan, M., Pang, R., Le, Q.V. (2020). Efficientdet: Scalable and efficient object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, pp. 10781-10790. https://doi.org/10.1109/CVPR42600.2020.01079