Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 5565561, 13 pages
https://doi.org/10.1155/2021/5565561

Research Article

Deep Transfer Learning Based Multiway Feature Pyramid Network for Object Detection in Images

Parvinder Kaur,[1] **Baljit Singh Khehra**,[2] **and Amar Partap Singh Pharwaha**[3]

[1] *Research Scholar, IKG PTU, Jalandhar, Punjab, India*
[2] *Department of CSE, BBSBEC Fatehgarh Sahib, Fatehgarh Sahib, India*
[3] *Department of ECE, SLIET Longowal, Longowal, India*

Correspondence should be addressed to Baljit Singh Khehra; baljit.singh@bbsbec.ac.in

Received 20 January 2021; Revised 23 March 2021; Accepted 3 April 2021; Published 19 April 2021

Academic Editor: Vijay Kumar

Object detection is being widely used in many fields, and therefore, the demand for more accurate and fast methods for object detection is also increasing. In this paper, we propose a method for object detection in digital images that is more accurate and faster. The proposed model is based on Single-Stage Multibox Detector (SSD) architecture. This method creates many anchor boxes of various aspect ratios based on the backbone network and multiscale feature network and calculates the classes and balances of the anchor boxes to detect objects at various scales. Instead of the VGG16-based deep transfer learning model in SSD, we have used a more efficient base network, i.e., EfficientNet. Detection of objects of different sizes is still an inspiring task. We have used Multiway Feature Pyramid Network (MFPN) to solve this problem. The input to the base network is given to MFPN, and then, the fused features are given to bounding box prediction and class prediction networks. Softer-NMS is applied instead of NMS in SSD to reduce the number of bounding boxes gently. The proposed method is validated on MSCOCO 2017, PASCAL VOC 2007, and PASCAL VOC 2012 datasets and compared to existing state-of-the-art techniques. Our method shows better detection quality in terms of mean Average Precision (mAP).

1. Introduction

Object detection is flouted into an extensive room of enterprises, with uses ranging from security to efficacy in the working environments. One very simple application can be locating the lost keys in a messy room. Other applications are surveillance, unmanned vehicles, counting the number of people in a scene, filtering, salacious images on the Internet, detecting abnormalities in scenes such as bombs, real-time vehicle detection in metro cities, machine investigation, image retrieval, face detection, pedestrian detection, activity recognition, human-computer interaction, service robots, and many more [1]. The beginning of the last decade was very lucky for deep learning due to the increased computational speed of GPU and the availability of extremely large datasets that contain millions of labeled data. These two proved booms to deep learning and object detection, and a series of object detection and localization methods started [2]. Overfeat [3] was proposed by Sermanet et al. in 2014. It used a single convolution neural network to perform classification, detection, and localization of objects in images. It also emphasizes on the concept that avoiding the training of background allows the network to focus on positive classes merely. However, in this method, they were not backpropagating through the whole network. R-CNN (Region with CNN features) [4] was proposed by Girshick et al. in 2014. It was an excellent achievement in the field of object detection. It combined the concept of region proposal with CNN. Selective search was used to extract 2000 regions from the image, and these regions were called region proposals. Support Vector Machine (SVM) was used for detection of objects. It gave 30 percent better performance over the existing methods. However, still this algorithm takes a large amount of time to train the network. Erhan et al. in 2014 proposed a saliency-based CNN for object detection that could handle the detection of multiple instances of the same object [5]. Spatial Pooling Pyramid network (SPPnet) [6] designed by He et al. speeded up R- CNN by avoiding repeated computation of convolution features. It also eliminated the requirement of the fixed size input image for CNN. Fast R-CNN [7], proposed by the same author of R-CNN in 2015, tried to solve some of the limitations of R-CNN. It is fast in the sense that instead of feeding 2000 region proposals to CNN every time, only one convolution operation is done once for each image to produce a feature map from it. Although Fast R-CNN and SPPnet made the detection networks faster but still the region proposal was a bottleneck. Faster R-CNN [8] was introduced by Ren et al. in 2015. This method replaced the slow selective search procedure for region proposal with a new concept called Region Proposal Network (RPN). Faster R-CNN was dramatically faster than its previous versions, so it could be used for object detection in real time. All previous object detection methods look into high probability regions of the images that may contain objects, and the training is a two-stage process, but You Look Only Once (YOLO) [9] uses a single CNN to generate class probabilities and bounding box predictions direct from the input image in only one pass making it faster than all previous methods. However, one major drawback of YOLO is that it fails to detect small-scale objects due to its spatial restrictions. Shrivastava et al. in 2016 proposed an online hard example mining (OHEM) [10] method for training region-based CNN for object detection. OHEM automatically selects difficult examples and excludes numerous heuristics and hyperparameters to streamline the training process of R-CNN. Liu et al. in 2016 proposed the Single-Shot Multibox detector (SSD) [11] that speeds up the system by removing the necessity of the region proposal network. It performs both detection and localization using small convolution filters. It achieves a better balance between speed and accuracy. Dai et al. in 2016 introduced Region-based Fully Convolutional Networks (R-FCN) [12]. This detector is fully convolutional with nearly all calculations shared with the whole input image. Recently, transfer learning models have been adopted to increase the efficiency of CNN models. All fully CNN image classifier-based networks such as Resnet [7, 13] and VGG net could be adopted by R-FCN for object detection automatically. YOLOv2 [14] was published by Redmon and Farhadi in 2017, introducing some improvements in YOLOv1. When compared to F-RCNN, it was noticed that YOLOv1 made many noticeable localization errors. Therefore, they applied batch normalization to all convolution layers. The method also increases the image resolution for object detection. They used anchor boxes instead of fully connected layers to predict bounding boxes. However, there is a little reduction in accuracy with the use of anchor boxes. YOLOv2 can work with a variety of sizes, attaining a good balance between speed and accuracy. RetinaNet [15] was proposed by Lin et al. in 2017. The model addresses the issue of the foreground-background class imbalance problem faced during the training process of single-short detectors. Single-short detectors are less accurate than two-stage detectors due to this problem. RetinaNet solves this problem by using a loss function named as focal loss which castigates simply classified examples, i.e., background examples.

The main contributions of this paper are as follows:

(a) In this paper, we propose a method for object detection in digital images based on Single-Stage Multibox Detector (SSD) architecture.

(b) This proposed model creates many anchor boxes of various aspect ratios based on the backbone network and proposed multidirectional feature pyramid network (MFPN) and calculates the classes and balances of the anchor boxes to detect objects at various scales.

(c) Instead of the VGG16-based deep transfer learning model in SSD, we have used a more efficient base network, i.e., EfficientNet. Softer-NMS is applied instead of NMS in SSD to reduce the number of bounding boxes gently.

The remaining paper is organized as follows. Section 2 discusses the related work. The proposed model is presented in Section 3. Experimental results are presented in Section 4. Section 5 concludes the paper.

2. Related Work

Modern object detectors can be classified as one-stage detectors and two-stage detectors. YOLO and SSD are state-of-the-art one-stage object detectors. The most promising two-stage detector is Faster R-CNN. Two-stage detectors are more accurate than single-stage detectors in terms of mean Average Precision (mAP). However, one-stage detectors are fast enough to make real-time object detection possible.

Figure 1 depicts the common architecture of modern object detectors. The base network plays an important role both in one-stage and two-stage detectors. 80% of the total time is consumed on the base network of SSD, concluding that, with a faster and more precise base network, SSD can perform better. Some new base networks like ResNet [16] and ResNeXt [17] have shown better performance and replaced Alex Net [18] and VGG Net [19] in recent object detection models. Every new BaseNet is increasing the number of layers to get better results. However, the problem of vanishing gradients [20] hinders the improvement in the performance. Moreover, the networks with a larger number of layers are difficult to train. Going deeper is not the only solution. Scaling can be done along the dimensions. EfficientNet [21] has solved this problem by introducing the concept of compound scaling. To get better accuracy, scaling is required along the dimensions, i.e., width, depth, and resolution. In EfficientNet, the resolution is scaled by 15%, depth by 20%, and width by 10%. The architecture of EfficientNet is described in Table 1 given below.

The architecture consists of 7 Inverted Residual Blocks (IRB), also called MBConv Blocks [22]. In this, 1×1

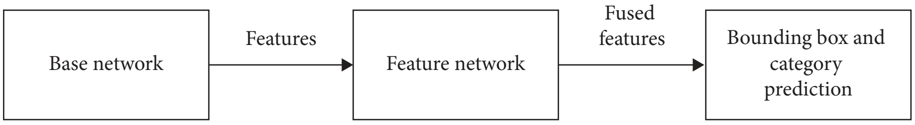


Figure 1: Architecture of the object detection model.

Table 1: Architecture of EfficientNet.

Level x	Convolution function F_X	Resolution $H_X \times W_X$	No. of channels C_X	No. of layers L_X
1	Convol 3×3	224×224	32	1
2	k3×3, IRB1	112×112	16	1
3	k3×3, IRB6	112×112	24	2
4	k3×3, IRB6	56×56	40	2
5	k3×3, IRB6	28×28	80	3
6	k3×3, IRB6	14×14	112	3
7	k3×3, IRB6	14×14	192	4
8	k3×3, IRB6	7×7	320	1
9	Convol 1×1, pooling, FC	7×7	1280	1

convolution is applied first, and then, depth-wise convolution of 3×3 is applied to decrease the number of attributes. In the last 1×1, convolution is applied to decrease the number of channels. In previous works, equal importance is given to the channels in the feature map received from the convolution layer. However, it is not possible that all channels are of the same importance. Therefore, to differentiate the features, some weight is assigned according to their significance. EfficientNet uses squeeze and excitation methods to treat the channels according to their importance by assigning some weights to them. Self-learning of weights is done by CNN.

Second, a building block of the object detector is the feature network that takes input features from the base network and outputs the fused features by considering the most salient features. Feature Pyramid Network (FPN) is used by many object detectors for this purpose. FPN is good at combining features at different scales. RetinaNet [15], PANet [23], and NAS-FPN [24] used FPN for feature fusion. They all simply add up the features by considering them equally important without considering their influence [7]. Therefore, there is a need to design a FPN that can take into consideration the multiscale features. We have tried to improve multiscale feature fusion by proposing a different approach that is based on the multidirectional feature pyramid. We will explain this architecture in detail in the next section.

Both one-stage and two-stage detectors use the concept of anchor boxes. In the $x \times y$ feature map, for each location, we get n anchor boxes with different aspect ratios and size. In YOLO, we get 98 anchor boxes, and in SSD, we get 8732 bounding boxes, which is much larger in number compared to YOLO. Out of the predicted bounding boxes by regression, which one is most appropriate and accurate in terms of intersection over union (IoU), can be determined by Nonmaximum Suppression (NMS). First, it chooses the bounding box with the best probability value. In the next step, it compares its IoU with other boxes. It eliminates the boxes with less and finds the best one from the predicted bounding boxes. It eliminates the bounding boxes with IoU greater than 50 percent. The process is repeated until there is no further elimination. NMS avidly agrees to select the highest-scored bounding box and eliminates all other bounding IoU which are greater than a specified threshold value. Nevertheless, accurate object location sometimes cannot be determined by high classification values and can lead to object localization disappointments. Many variations of NMS such as Regression-NMS [25, 26], Soft-NMS [27, 28], and Softer-NMS [29, 30] have been implemented in literature. Soft-NMS animatedly decreases the score value on the basis of the recently computed NMS. Whenever greater overlap is detected, it associates a higher score and so a greater chance of elimination. Softer-NMS tries to remove two problems in NMS: (a) the first problem arises when all bounding boxes for an object are imprecise in any of the coordinates [31] and (b) the second problem arises when a bounding box with the precise location is assigned a low confidence score [32].

3. Proposed Method

In this section, first we will discuss the reason why we have changed the base network of SSD. Then, we will explain the architecture of the proposed Feature Pyramid Network, i.e., MFPN and the mathematics behind it. In the last, we will explain the NMS method applied in our work. Figure 2 shows the detailed architecture of the proposed method.

3.1. Base Network. Images of size 224×224 were used to train early CNNs. Modern CNNs are trained on 480×480 image size. In our proposed work, we have trained the network with an image resolution of 1024×1024. An increase in resolution allows the system to extract detailed features. One more thing is that high resolution images must need CNNs with more layers, i.e., depth scaling. The reason behind deeper networks is that bigger receptive fields can extract alike features that include more pixels in high resolution images. Width of a network (number of channels) can also be increased to acquire the fine-grained features in an image. That is why we have replaced the traditional VGG net with EfficientNet. EfficientNet increases system speed and accuracy by using the concept of compound scaling (width, depth, and resolution) as discussed in the previous section.

3.2. Multiway Feature Pyramid Network (MFPN). In SSD, after the feature extraction phase, we obtain a feature layer of size $x \times y$ with n channels (8×8 or 12×12 or larger). Following which a 3×3 convolution is performed on $x \times y \times n$ feature layer to get fused features from multiple scales. We have replaced these extrafeature layers with Multiway Feature Pyramid Network (MFPN) that groups features at different resolutions. MFPN allows the detected features to flow in multiple directions to get better fused features. Features detected at various resolutions do not always pay the same weightage to output of the system. Extra weights are assigned to each input layer so that the network can learn the significance of each filter fusion process. Instead of traditional convolutions, we have applied depth-wise detachable convolution. Steps to fuse low-level features with high-level features are given below:

(I) Nodes with one input edge do not need any feature fusion. Therefore, such nodes are removed.

(II) If the input and output nodes are at the same level, then an extra edge is added between them.

(III) The two-way path is built so that it can be repeated multiple times to get better feature fusion.

(IV) Apply weighted fusion given as follows:

$$O = \sum_{m} \frac{w_m}{\in + \sum_n w_n} \cdot I_m, \tag{1}$$

where I_m represents the input features at level m and w_m is the learnable weight input features at level m. Value of $\in$ is a small random value near to and greater than zero.

(V) Integrate MFPN multiscale connections with weighted fusion as given below:

$$F_{\text{inter}}^{n} = \text{Convol}\left(\frac{w_1 \cdot F_{\text{in}}^{n} + w_2 \cdot \text{Resize}\left(F_{\text{in}}^{n+1}\right)}{w_1 + w_2 + \in}\right), \tag{2}$$

$$F_{\text{out}}^{n} = \text{Convol}\left(\frac{w_1' \cdot F_{\text{in}}^{n} + w_2' \cdot F_{\text{inter}}^{n} + w_3' \cdot \text{Resize}\left(F_{\text{out}}^{n-1}\right)}{w_1' + w_1' + w_1' + \in}\right), \tag{3}$$

where F_{inter}^{n} represents features at intermediate level n on the top-down path of MFPN and F_{out}^{n} represents the output features at level n on the bottom-up path of the MFPN.

Input features from the feature layers ($F2$, $F3$, $F4$, $F5$, $F6$, and $F7$) of EfficientNet are fed to MFPN for multiway and multiscale feature fusion. The output from MFPN is given to the classification and bounding box regression head to eliminate the number of detected bounding boxes' softer-NMS [31] is used instead of normal NMS in SSD. Table 2 shows the configuration details of the proposed system. Our work is mainly inspired by Tan et al. [33].

4. Implementation Details and Experiments

The experiments including training are done on Ubuntu 18.04. The frameworks used are TensorFlow 2.0 and OpenCv 3.4.9. Hardware used is Dell Precision T3500 Workstation with Intel Xeon 5600 series processor, CUDA enabled TITAN XP GPU with 12 GB RAM. The experiments are carried out on the MSCOCO 2017 dataset. The performance is also tested on the PASCAL VOC 2007 test set. Stochastic Gradient Descent is used for optimization with momentum of 0.9 and weight decay of $4e-5$. The learning rate is increased steadily from 0.0 to 0.12 during very first epoch of each model's training period. Learning rate annealing is used after that to get the adaptive learning rate. After each convolution layer, batch normalization is used. Batch size used for training is 128, and number of epochs for each model is 300. The Swish activation function is used instead of ReLU due to its simplicity. The results are compared with state-of-the-art techniques on the basis of mean Average Precision (mAP). Scaling the network along all dimensions-width, depth, and resolution leads to improved performance. EfficientNet uses this concept of compound scaling and outperforms AlexNet, VGG16, and ResNet50 for object detection in terms of mAP, as shown in Table 3.

Table 4 shows the various dimensions and parameters of the proposed system and the state-of-the-art techniques. The number of parameters of the proposed system is less when compared with better performance in terms of mAP.

Table 5 shows the overall proposed system's comparison with state-of-the-art techniques, and Figures 3 and 4 show the visual results. Simply replacing the base network gives 7% improvement in mAP and replacing FPN with the proposed feature fusion network gives an increase of 4%. Using Softer-NMS gives an improvement of 3%. Applying augmentation techniques gives 2% improvement.

Figure 4 depicts the qualitative performance evaluation of the state-of-the-art techniques and the proposed

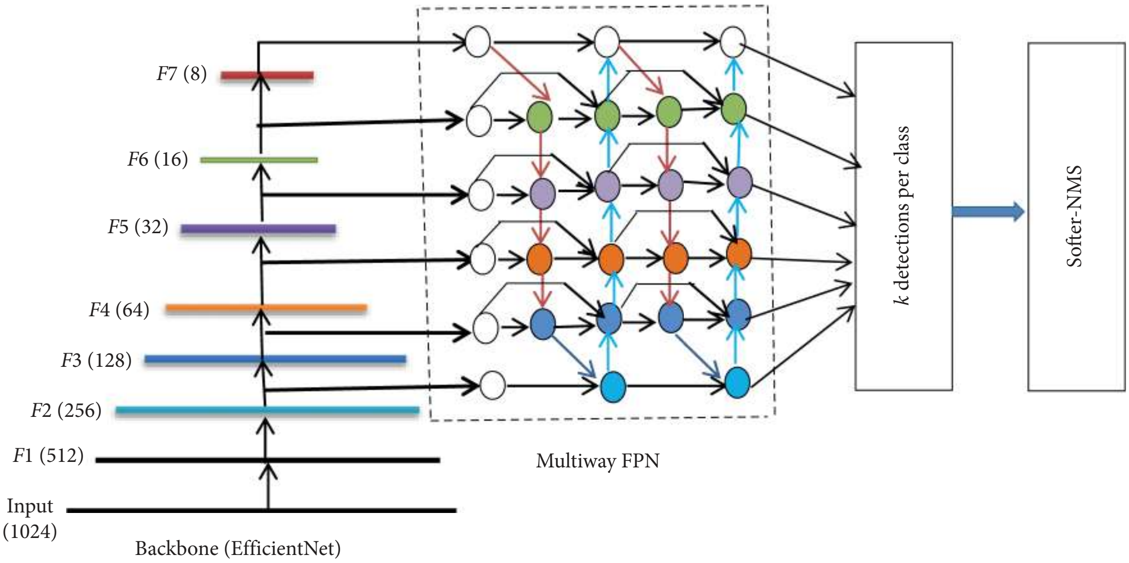


Figure 2: Architecture of the proposed system.

Table 2: Configuration details of the proposed system.

Input size	No. of channels (MFPN)	No. of layers (MFPN)	No. of layers' bounding box and classification head	Suppression method
1024	224	7	4	Softer-NMS

Table 3: Comparison of backbone networks with different feature fusion architectures on the MSCOCO test set.

Backbone network	Feature fusion network	mAP
AlexNet	FPN	33.06
VGG16	FPN	38.41
ResNet50	FPN	44.98
EfficientNet B4	FPN	**45.32**
EfficientNet B4	MFPN (proposed)	**49.21**

Table 4: Model characteristics.

Model	Backbone	Image resolution	Parameters (M)
SSD	Mobilenet V2	224 × 224	3.22
Faster -RCNN	Inception_ResNet V2	600 × 1024	13.3
YOLOv3	Darknet	416 × 416	65
Proposed	EfficientNet B4	1024 × 1024	21

Table 5: Performance evaluation and comparison in terms of average precision (AP).

Model	Backbone	mAP (PASCAL VOC 2007)	mAP (PASCAL VOC 2012)	mAP MSCOCO
SSD	Mobilenet V2	68.02	71.02	42.07
Faster-RCNN	Inception_ResNet V2	70.57	73.18	46.79
YOLOv3	Darknet	71.24	68.90	50.73
YOLOv3	VGG16	62.53	64.77	50.82
YOLOv3	AlexNet	56.90	58.02	49.84
Proposed + S-NMS	EfficientNet	73.13	74.85	52.62
Proposed with augmentation	EfficientNet	76.01	77.43	54.01

system. The other algorithms are not detecting all instances of objects, i.e., the number of false negatives for the three algorithms is more than the proposed algorithm, that is a great improvement. Figures 5–10 show the precision call curves for the state-of-the-art and proposed systems.

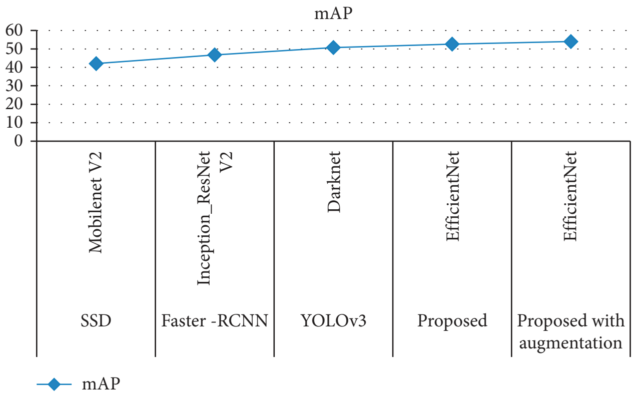
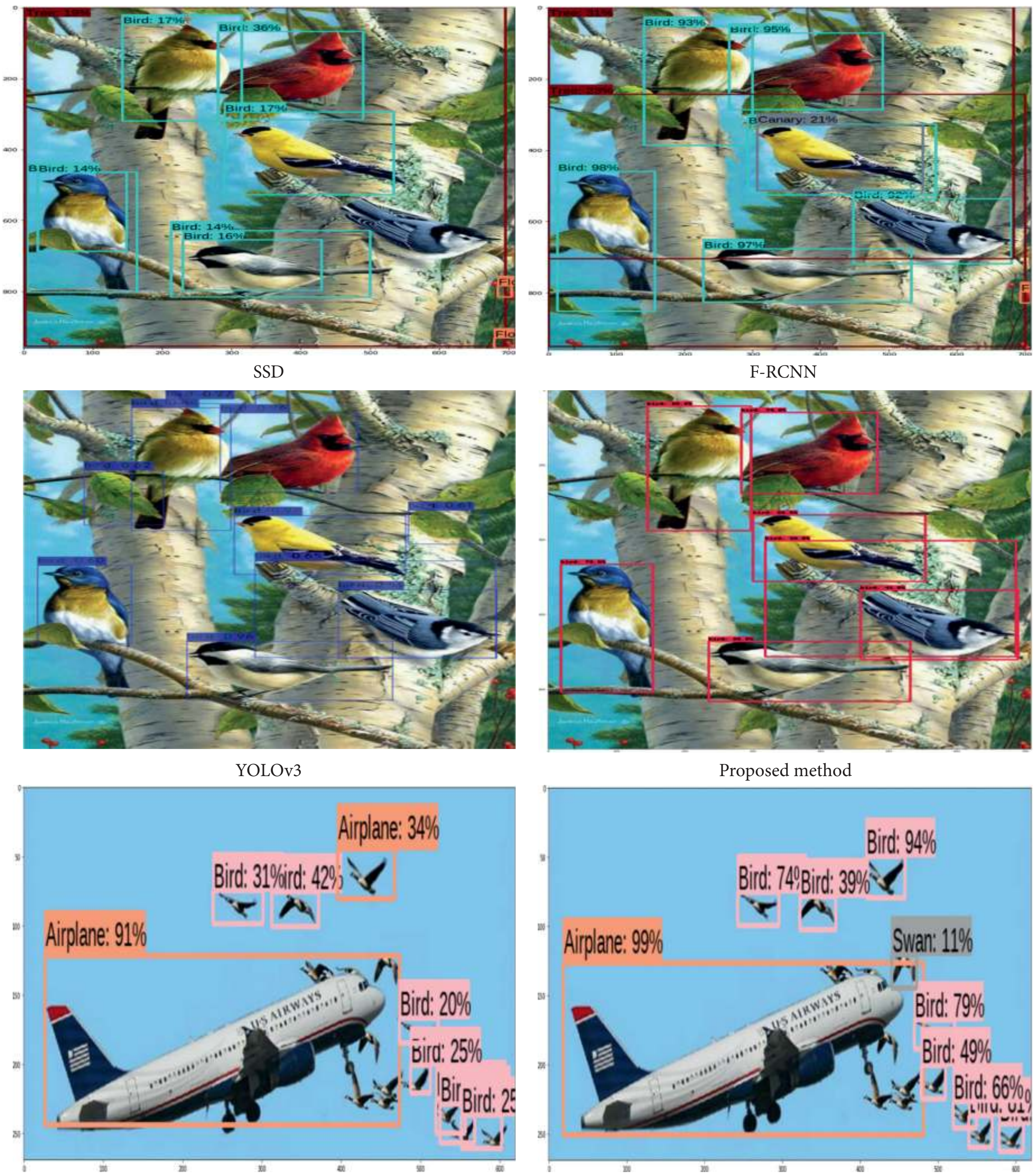


Figure 3: Comparison with state-of-the-art techniques on MSCOCO 2017.



SSD

F-RCNN

YOLOv3

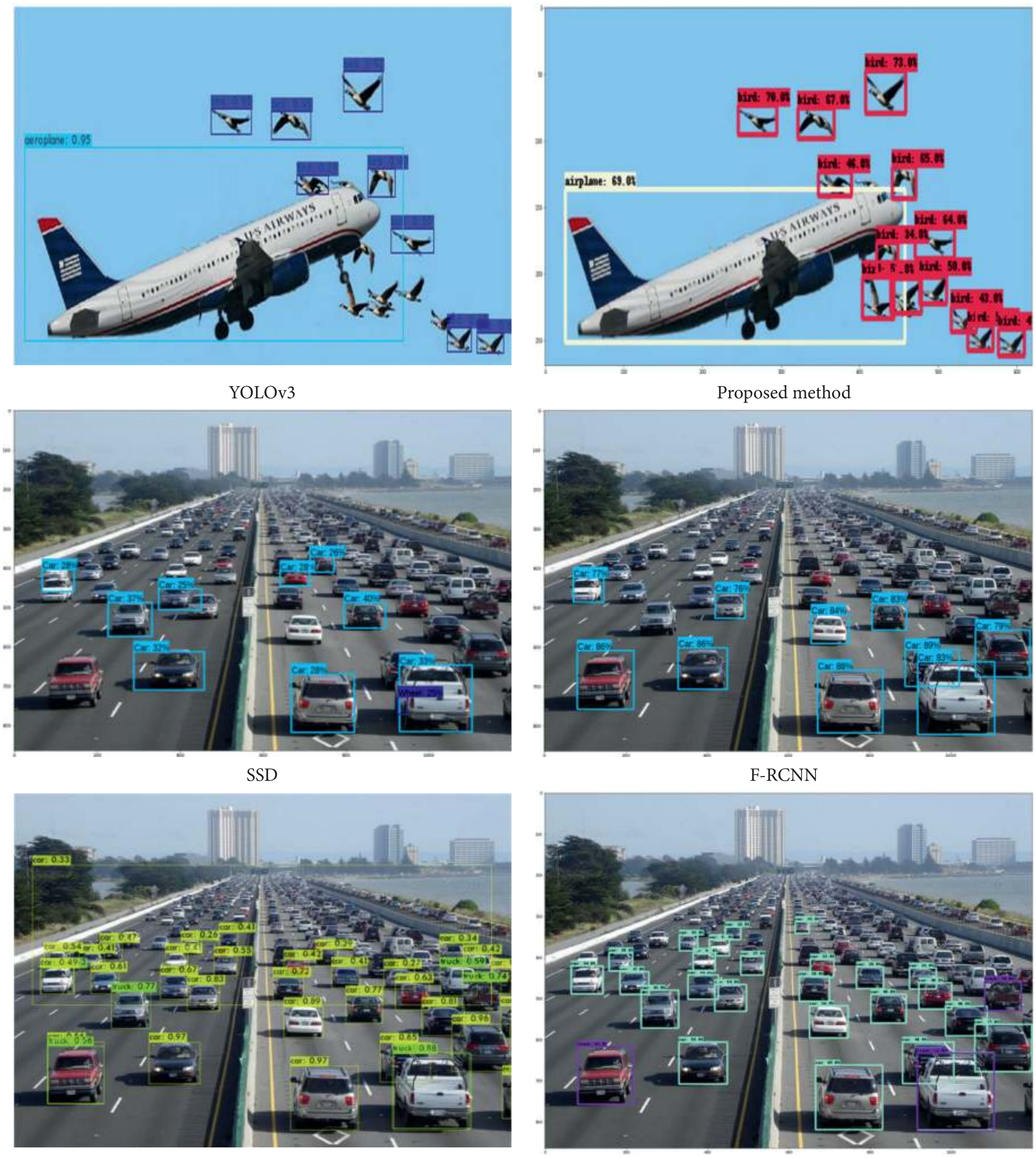
Proposed method

SSD

F-RCNN

(a)

Figure 4: Continued.



YOLOv3

Proposed method

SSD

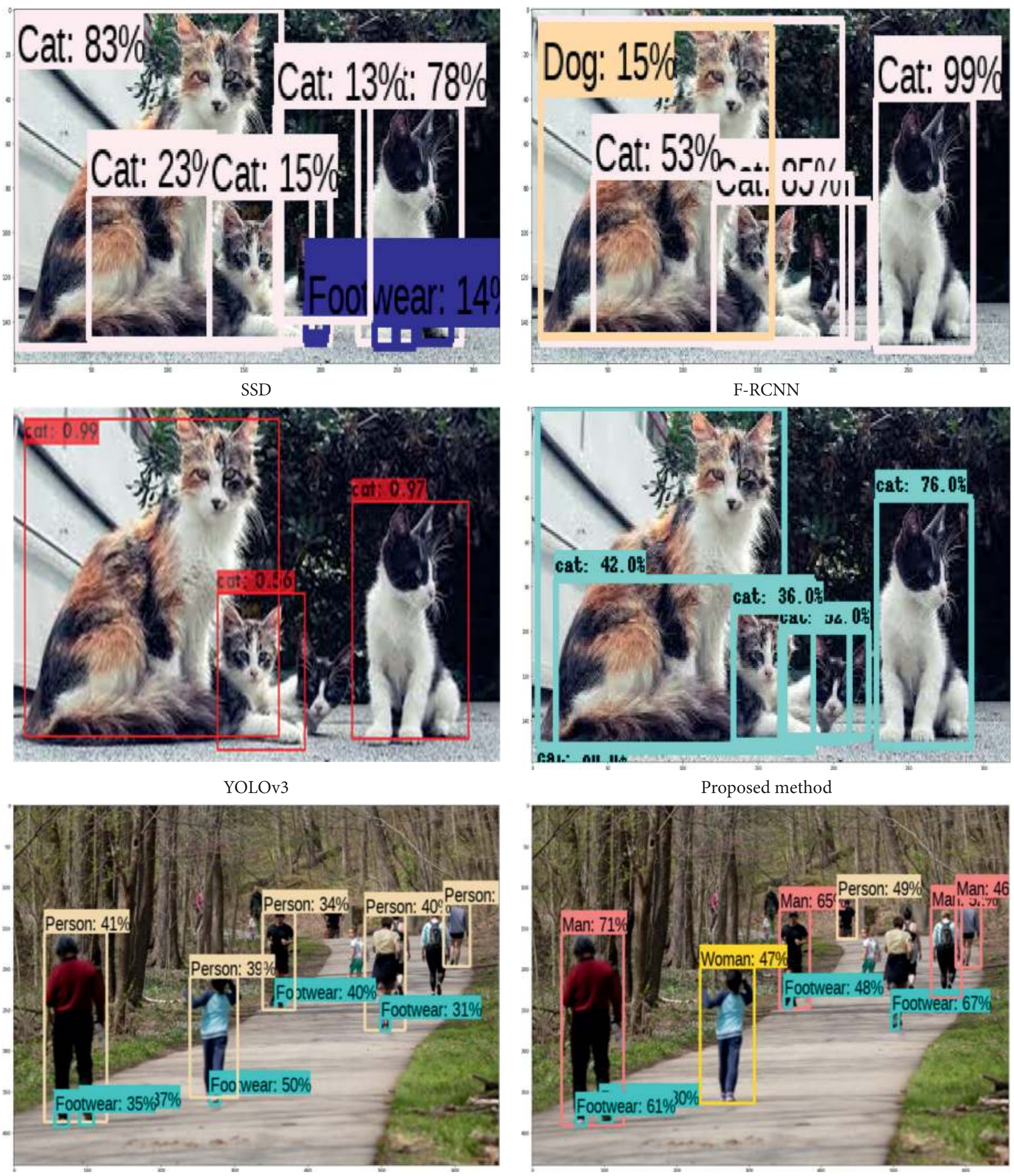
F-RCNN

YOLOv3

Proposed method

(b)

Figure 4: Continued.



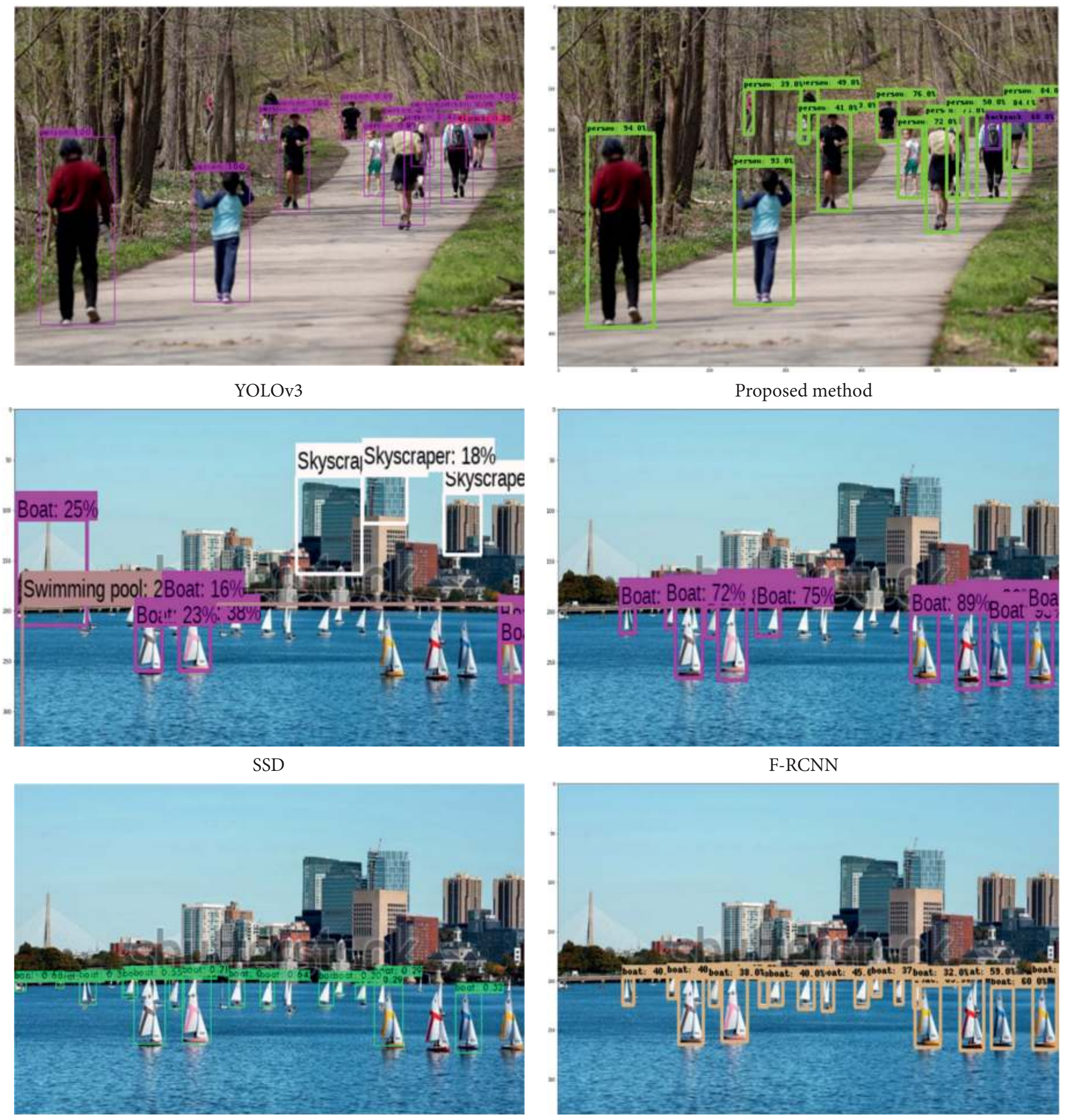
SSD F-RCNN

YOLOv3 Proposed method

SSD F-RCNN

(c)

Figure 4: Continued.



YOLOv3

Proposed method

SSD

F-RCNN

YOLOv3

Proposed method

(d)

Figure 4: Comparison of object detection results.

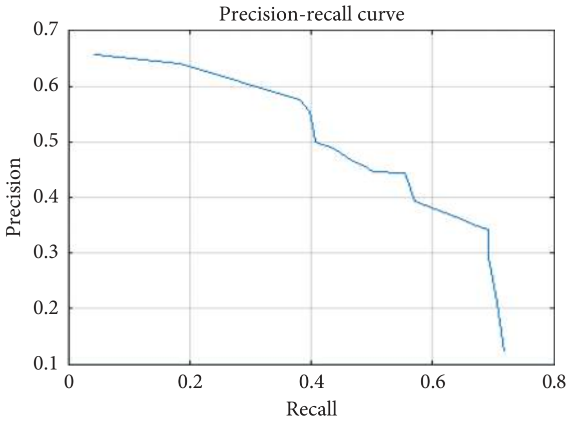


Figure 5: Precision-recall curve for SSD.

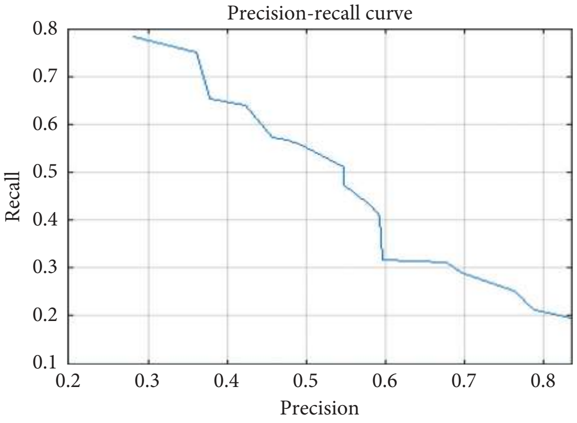


Figure 6: Precision-recall curve for faster-RCNN.

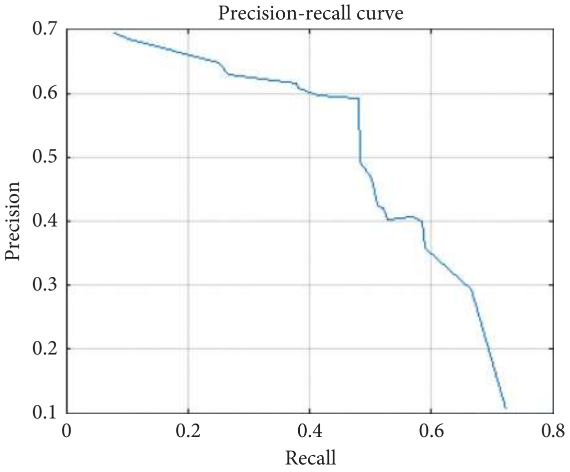


Figure 7: Precision-recall curve for YOLOv3.

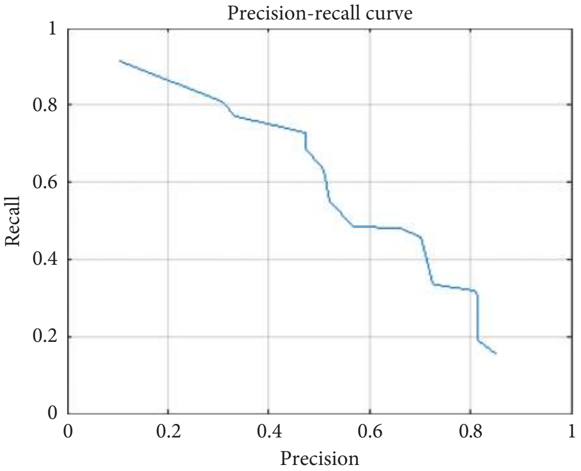


Figure 8: Precision-recall curve for the proposed system without augmentation.

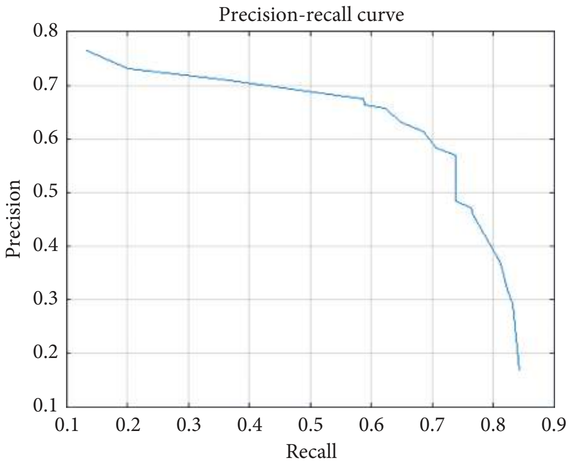


Figure 9: Precision-recall curve for the proposed system with augmentation.

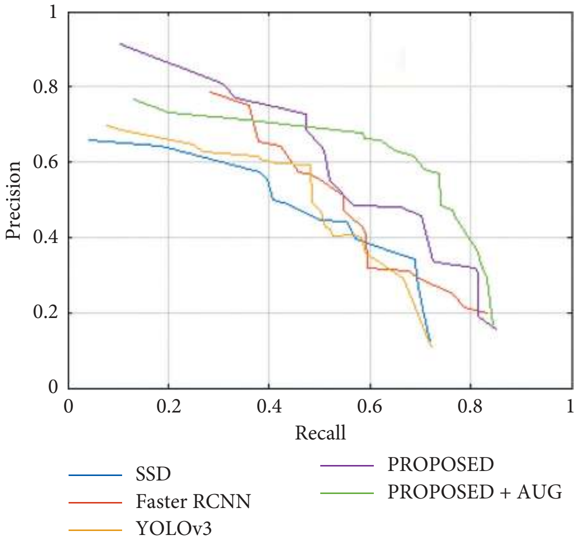


Figure 10: Precision-recall curves comparison (MSCOCO).

5. Conclusion and Future Scope

Images at higher resolution can be more helpful for the detection of small objects. Keeping this idea in mind, an input image of 1024 × 1024 is taken as input. EfficientNet is used as a backbone to provide a combination of depth, width, and resolution scaling. To fuse high-level features with lower-level features at different scales, Multiway Feature Pyramid Network is used with weighted fusion. The system is further improved with Softer-NMS. The quantitative results show an improvement of 4% in mean Average Precision (mAP) on the MSCOCO dataset. Subjective results show that the number of false negatives in the proposed technique is less than the number of false negatives in state-of-the-art techniques.

In the future, we will try to enhance the results by using the ensembling of the deep transfer learning model. Additionally, the proposed model can be tested for other kinds of datasets.

Data Availability

Data will be made available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. R. Pathak, M. Pandey, and S. Rautaray, "Application of deep learning for object detection," *Procedia Computer Science*, vol. 132, pp. 1706–1717, 2018.

[2] W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: a survey," *International Journal of Computer Vision and Pattern Recognition*, vol. 128, pp. 261–318, 2020.

[3] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: integrated recognition, localization and

detection using convolutional networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, Canada, April 2014.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, June 2014.

[5] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CPVR)*, Columbus, OH, USA, June 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Glasgow, UK, August 2015.

[7] M. Kaur and D. Singh, "Multi-modality medical image fusion technique using multi-objective differential evolution based deep neural networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 1–11, 2020.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "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, 2017.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[10] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 761–769, Las Vegas, NV, USA, June 2016.

[11] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single Shot MultiBox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, Netherlands, October 2016.

[12] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: object detection via region-based fully convolutional networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 379–387, Barcelona, Spain, December 2016.

[13] R. Wang, H. Yu, G. Wang, G. Zhang, and W. Wang, "Study on the dynamic and static characteristics of gas static thrust bearing with micro-hole restrictors," *International Journal of Hydromechatronics*, vol. 2, no. 3, pp. 189–202, 2019.

[14] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 6517–6525, Honolulu, HI, USA, July 2017.

[15] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, Venice, Italy, October 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[17] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, Honolulu, HI, USA, July 2017.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, vol. 1, pp. 1097–1105, Siem Reap, Cambodia, December 2012.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[20] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *Proceedings of the 3rd Asian Conference on Pattern Recognition (ACPR)*, pp. 730–734, Kuala Lumpur, Malaysia, November 2015.

[21] M. Tan and Q. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," in *Proceedings of the 2019 International Conference on Machine Learning (ICML)*, Long Beach, CA, USA, June 2019.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018.

[23] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, Salt Lake City, UT, USA, June 2018.

[24] G. Ghiasi, T. Lin, and Q. V. Le, "NAS-FPN: learning scalable feature pyramid architecture for object detection," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7029–7038, Long Beach, CA, USA, June 2019.

[25] S. Qiu, G. Wen, Z. Deng, J. Liu, and Y. Fan, "Accurate non-maximum suppression for object detection in high-resolution remote sensing images," *Remote Sensing Letters*, vol. 9, no. 3, pp. 238–247, 2017.

[26] S. Ghosh, P. Shivakumara, P. Roy, U. Pal, and T. Lu, "Graphology based handwritten character analysis for human behaviour identification," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 1, pp. 55–65, 2020.

[27] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS - improving object detection with one line of code," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5562–5570, Venice, Italy, October 2017.

[28] B. Gupta, M. Tiwari, and S. Singh Lamba, "Visibility improvement and mass segmentation of mammogram images using quantile separated histogram equalisation with local contrast enhancement," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 2, pp. 73–79, 2019.

[29] Y. He, X. Zhang, K. Kitani, and M. Savvides, "Softer-NMS: rethinking bounding box regression for accurate object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, June 2019.

[30] H. S. Basavegowda and G. Dagnew, "Deep learning approach for microarray cancer data classification," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 1, pp. 22–33, 2020.

[31] S. Osterland and J. Weber, "Analytical analysis of single-stage pressure relief valves," *International Journal of Hydromechatronics*, vol. 2, no. 1, pp. 32–53, 2019.

[32] T. Wiens, "Engine speed reduction for hydraulic machinery using predictive algorithms," *International Journal of Hydromechatronics*, vol. 2, no. 1, pp. 16–31, 2019.

[33] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: scalable and efficient object detection," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10778–10787, Seattle, WA, USA, June 2020.