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

Traitement du Signal
Vol. 38, No. 2, April, 2021, pp. 481-494
Journal homepage: http://iieta.org/journals/ts

A Region-Based Efficient Network for Accurate Object Detection

Yurong Guan[1], Muhammad Aamir[1], Zhihua Hu[1*], Waheed Ahmed Abro[1], Ziaur Rahman[1], Zaheer Ahmed Dayo[1], Shakeel Akram[2]

[1] Department of Computer, Huanggang Normal University, Huanggang 438000, China
[2] College of Electrical Engineering, Sichuan University, Chengdu 610000, China

Corresponding Author Email: huzhihua@hgnu.edu.cn

https://doi.org/10.18280/ts.380228

Received: 30 December 2020
Accepted: 27 February 2021

Keywords:
object detection, object classification, proposal generation, proposal refinement, proposal classification

ABSTRACT

Object detection in images is an important task in image processing and computer vision. Many approaches are available for object detection. For example, there are numerous algorithms for object positioning and classification in images. However, the current methods perform poorly and lack experimental verification. Thus, it is a fascinating and challenging issue to position and classify image objects. Drawing on the recent advances in image object detection, this paper develops a region-based efficient network for accurate object detection in images. To improve the overall detection performance, image object detection was treated as a twofold problem, involving object proposal generation and object classification. First, a framework was designed to generate high-quality, class-independent, accurate proposals. Then, these proposals, together with their input images, were imported to our network to learn convolutional features. To boost detection efficiency, the number of proposals was reduced by a network refinement module, leaving only a few eligible candidate proposals. After that, the refined candidate proposals were loaded into the detection module to classify the objects. The proposed model was tested on the test set of the famous PASCAL Visual Object Classes Challenge 2007 (VOC2007). The results clearly demonstrate that our model achieved robust overall detection efficiency over existing approaches using fewer or more proposals, in terms of recall, mean average best overlap (MABO), and mean average precision (mAP).

1. INTRODUCTION

Object detection a prominent problem in machine vision and computer vision. In many applications, object detection devices are a crucial component. Object detection has been widely used for applications and systems of environmental reasoning [1-5]. Therefore, rigorous research in this field is always active and essential.

The goal of object detection is to locate and classify different objects in an image automatically. The detection process could be affected by various factors, such as low image quality, noise, and background interference. Therefore, it is challenging to realize state-of-the-art speed and quality in object detection. To overcome this challenge, numerous approaches have been proposed to detect objects in images efficiently [6, 7].

Currently, object detection systems mostly comprise of two stages: positioning of objects in an image, and classifying these positions. Rather than single stage optimization, the two stages must be improved simultaneously to achieve desired detection performance.

The generation of object proposals draws a bounding box on the regions containing the objects of interest in the target image. Focusing only on the candidate proposals that are believed to contain the desired objects, this technique aims to reduce the computing load of solving the pixel-by-pixel similarity among objects in the entire image. An ideal object proposal generator should achieve a high recall with a limited number of proposals.

For two-stage object detection tasks, the object proposals can be generated by Rantalankila's method [8], geodesic object proposal (GOP) [9], Rahtu's method [10], image window objectness measurement (Objectness) [11], binarized normed gradients (BING) [12], randomized prim's algorithm [13], selective search [14], cascade support vector machine (CSVM) [15], learning to propose objects (LPO) [16], edge boxes [17], multiscale combinatorial grouping (MCG) [18], Endres' method [19], DeepBox [20], regional proposal network (RPN), DeepMask [21], SharpMask [22], and constrained parametric min-cuts [23]. However, these techniques face some noticeable limits in object detection, namely, low positioning accuracy, lack of scoring mechanism, high computing cost, low precision, class dependence, etc.

With the development of deep learning (DL), the classification accuracy of detection systems has been constantly improving. Lots of efforts have been invested to advance DL frameworks for robust object classification. Convolutional neural networks (CNNs) are powerful frameworks for the classification stage in object detection. The most representative CNNs include AlexNet [24], ResNet [25], DenseNet [26], network in network [27], GoogleNet [28], VGGNet [29], and other variants. The classification performance of these networks has been optimized by numerous regularization methods [30, 31]. However, the CNNs face limitations in model size, computing cost, and memory consumption, and are redundant and dubious to particular proposal generation methods.

The current region-based or proposal-based object detectors include region-based CNN (R-CNN) [32], spatial pyramid pooling network (SPPnet) [33], Fast R-CNN [34], Faster R-CNN [35], region-based fully convolutional network (R-FCN) [36], Mask R-CNN [37], and Cascade R-CNN [38].

Being the earliest region-based detector, R-CNN generates top-down region proposals for the target image through selective search, and classifies the positions into different categories. Despite its simple structure, this network brings a high computing cost, because each region is executed separately, and a large disk space is occupied by multistage training. Furthermore, R-CNN is slow in test, and only compatible with fixed-size input.

Therefore, R-CNN was improved into SPPnet, which supports variable-size input and share computing. Like R-CNN, SPPnet also needs a large disk space for multistage training, and operates slowly in object detection.

Later, Ross Girshick improved R-CNN and SPPnet into Fast R-CNN for object detection. Fast R-CNN processes the entire image at once, while R-CNN processes the image region by region. This network outperforms R-CNN and SPPnet both in computing cost and memory. The training is sped up by reducing the number of forward computations. The main disadvantage of Fast R-CNN lies in the selective search for proposals from the image.

Next, Faster R-CNN was developed based on two modules: deep CNN (DCNN) and Fast R-CNN detector. The former is responsible for proposal generation, and the latter, proposal classification. The network overcomes the slow speed and high computing cost of earlier approaches, by replacing selective search with a powerful RPN for region extraction. The replacement ensures the detection accuracy with fewer proposals. Nonetheless, Faster R-CNN has one drawback: it takes a long time to reach convergence.

Furthermore, He et al. [37] developed the simple, intuitive, fast, accurate, and easy-to-use Mask R-CNN based on pixel-level image segmentation. As an extension of Faster R-CNN, Mask R-CNN is known for its excellent detection performance. But the reliability of the network comes at the cost of high memory demand, slow detection, and poor real-timeliness.

To realize robust object detection, Dai et al. [36] presented the R-FCN, which relies on an FCN to speed up the detection. Besides, the shared convolution is adopted to directly act on the whole image. Compared with previous approaches, the R-FCN can achieve a high accuracy at a moderate speed.

Moreover, the state-of-the-art two-stage R-CNN object detection systems was extended into the Cascade R-CNN, with the goal of optimizing detection performance. In fact, the idea of cascading can be applied to any two-stage object detector, ranging from R-CNN, Fast RNN, Faster R-CNN, to Mask R-CNN.

Apart from two-stage techniques, single-stage methods also play an important role in object detection. Researchers have worked hard to develop single-stage proposal-free techniques, which are relatively advantageous and applicable to various real-time applications. Such techniques include you only look once (YOLO) [39], and single shot detector (SSD) [40]. The single-stage techniques are faster but less accurate than regional-based CNNs.

The above analysis shows a lack of experimental evidences on object detection in images. Consequently, this paper presents a region-based efficient network (REN) for accurate object detection in natural images. Derived from the prior methods, the REN can effectively reduce the computing cost of object detection, and enhance the detection accuracy.

The remaining part of the manuscript is organized as follows: Section 2 reviews the related works; Section 3 elucidates the REN; Section 4 verifies the REN performance through experiments; Section 5 discusses the experimental results; Section 6 gives the conclusions, and talks about the future work.

2. LITERATURE REVIEW

Object detection in images is a fundamental problem of computer vision. The fast-evolving object detection techniques has brought enormous commercial values, and penetrated every industry. Numerous techniques are now available to achieve state-of-the-art detection performance. Recently, object detection has been extensively used in various fields, such as medicine [41], roads, building detection [42, 43], automatic detection of lane marking [44], face detection [45], and pedestrian detection [46].

Object detection mimics the visual sensing of our brains, which can detect, process, and interpret visual information efficiently. In fact, a large part of the human brain is dedicated to the processing of visual information. In contrast with human intelligence, systems could not sense or process information appropriately for several challenging factors: variations in viewpoints and perspectives, illumination, small object scale, deformation, occlusion, rotation, and high intra-class difference. Most objects in images are influenced by these factors, making them difficult to detect accurately and efficiently. Researchers are obliged to modify the object detection systems, and enhance their generalization ability.

This section intends to comprehensively assess the traditional DL-based object detection methods, including both single-stage and two-stage algorithms. As mentioned in the preceding section, the most notable two-stage region-based CNNs include R-CNN, SPPnet, Fast R-CNN, Faster R-CNN, R-FCN, Mask R-CNN, and Cascade R-CNN. All of them have been adopted widely to achieve desired performance in object detection. Experimental results show that these networks are widely recognized for their excellence in various detection tasks. However, these methods require heavy computations, a long runtime, and a large disk space.

To maintain a high detection accuracy, Kaya et al. [47] proposed a model based on proposals improved by convolutional contact features. Their model outperforms the earlier methods, but only applies to classes with distinctive contexts. Liu et al. [48] combined RPN with Fast R-CNN into a DL-based model, which yields high-quality performance in object detection. Yet, the model fails to achieve high detection accuracy and speed. Zhang et al. [49] put forward an object detection model named anchor free (AF) R-CNN based on feature fusion and attention mechanism. The model is more accurate than previous region-based approaches. However, it is computationally complex and not suitable for real-time object detection. The slow detection speed remains a big challenge for this model. Xiao et al. [50] introduced an efficient detection model that retains the legacy of region-based CNNs. The model integrates skip pooling with contextual information instead of RPN, thereby realizing improved performance. But it still lags in processing speed. Similarly, inside-outside net (ION) [51], Hypernet [52], and online hard example mining (OHEM) [53] algorithms fail to achieve good real-timeliness or high detection accuracy. None of these models does well in the detection of large datasets.

To speed up the detection process, one-stage algorithms like YOLO and SSD were launched to predict and classify object positions. The YOLO meshes the target image into multiple grids, and then performs positioning and classification. However, it flops on small object detection, due to the lack of low-level high-resolution information. Zhao et al. [54] mixed YOLO alternatives into a novel YOLO model that efficiently and rapidly detects objects. But the model only works well on non-graphics processing unit (GPU) devices.

To address the issues of YOLO, the SSD was introduced to detect objects, using both low- and high-level feature maps with high-resolution information. This model works excellently on small objects and achieves favorable accuracy. However, it is open to further improvements. Deng et al. [55] presented an SSD model based on feature fusion and spatial attention. Despite its fast detection speed, the model is incompetent for the detection of large images: the image size does not affect the detection precision, but slows down the detection speed. Thus, many scholars have developed variants of the SSD to improve overall precision [56-60].

In the past studies, both one- or two-stage detection methods broadly adopt anchors. The traditional approaches typically produce anchors with RPN. Later, the anchors are directly classified and regressed. The number and shape of anchors greatly affect the performance of object detection algorithms. Wang et al. [61] employed an anchor that derives the number of sparse, arbitrary shapes from image features, and used these shapes to cut the number of anchors and improve their shapes, while ensuring a reliable recall. Comparatively, two-stage algorithms are slower yet more accurate than one-stage algorithms.

In addition, Wan et al. [62] proposed a min-entropy latent model (MELM) for object positioning and classification, and verified its superiority over the state-of-the-art approaches. Based on dissimilarity coefficient, Arun et al. [63] presented a probabilistic learning model capable of dealing with fuzzy object positions, and demonstrated that the model is more accurate than prior approaches. Fang et al. [64] developed an object detection model that solves the non-convexity problem with a series of smoothed loss functions, and proved the good overall performance of the model, as well as its advantage in positioning. Based on semantic segmentation, Shen et al. [65] proposed an object detection model with a multi-tasking learning mechanism, and observed that the model attained competitive outcomes against other alternatives. Yang et al. [66] developed an image object positioning model, which, unlike the earlier approaches, is immune to local minimum trap. The model combines multiple instances of learning and bounding-box regression in a single network, resolving the absence of instance-level class labels. Tang et al. [67] designed an object detection model based on proposal cluster learning. The model significantly improves the detection performance, but fails to work on deformed non-rigid objects.

Most of the above approaches purely rely on image-level labels. Compared to other proposal generation methods, these approaches do not have bounding box labels. This significantly bottlenecks the positioning ability of these approaches. In contrast, our method yields more insights into the overall performance of object detection, and draws a distinction in empirical findings.

3. OUR METHOD

This paper proposes a useful model for object detection in images. Figure 1 illustrates the functional blocks of our model. Different stages of the model are discussed in the following subsections, including object proposal generation, and proposal refinement & classification.

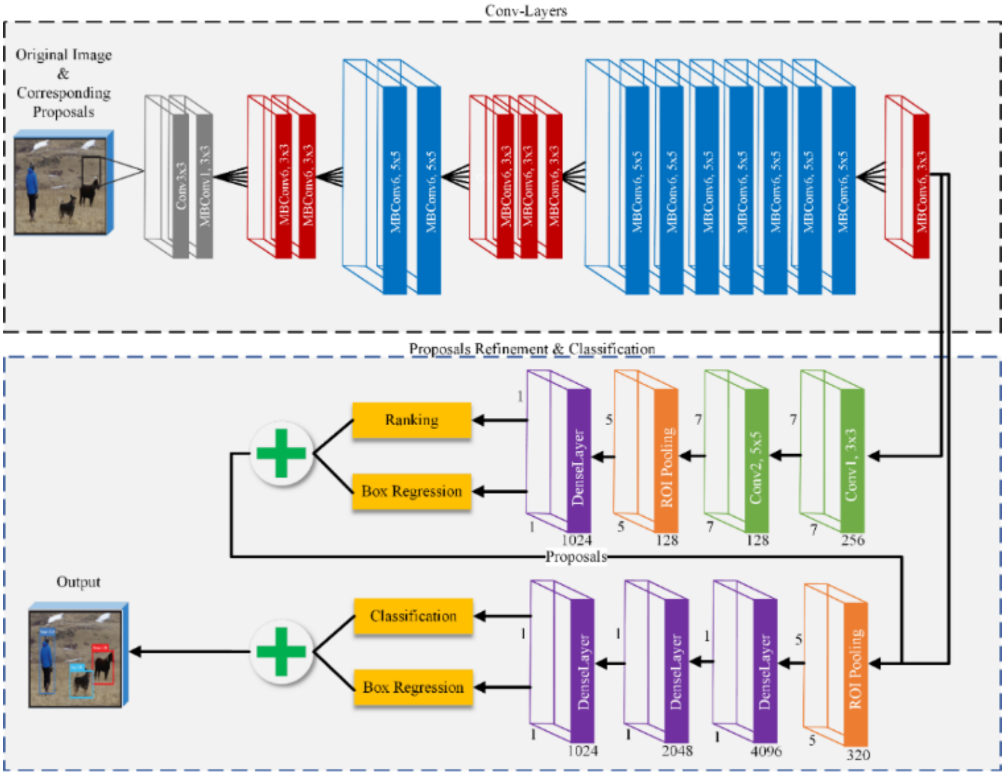

Figure 1. Framework of our model

3.1 Object proposal generation

The first stage of our model is to generate a few high-quality, class-independent proposals. Previous research has shown that a small set of proposals can immensely improve the performance of object detection. But the existing strategies are inadequate to produce a limited number of high-quality proposals.

To solve the problem, this paper first segments the target image into a set of initial regions, because segmentation could improve the effect of object detection. Compared to pixel regions with rich information, it is a good idea to draw object proposals from region-based features. Here, the set of initial regions is obtained by segmenting the image with the graph generation method proposed by Felzenszwalb and Huttenlocher [68]. This method is fast and accurate enough for our purpose. Each region thus acquired was considered as a cluster. Based on regional similarities, the adjacent regions were grouped from bottom to top by a cluster-based hierarchical strategy.

Firstly, the similarities between adjacent regions were calculated, and used to merge the most similar regions into one region. Then, the similarities between the merged regions were calculated, and used to combine the most similar ones into a region. The process of merging similar regions was executed iteratively until all the similar regions had been fused into a single region to form an image.

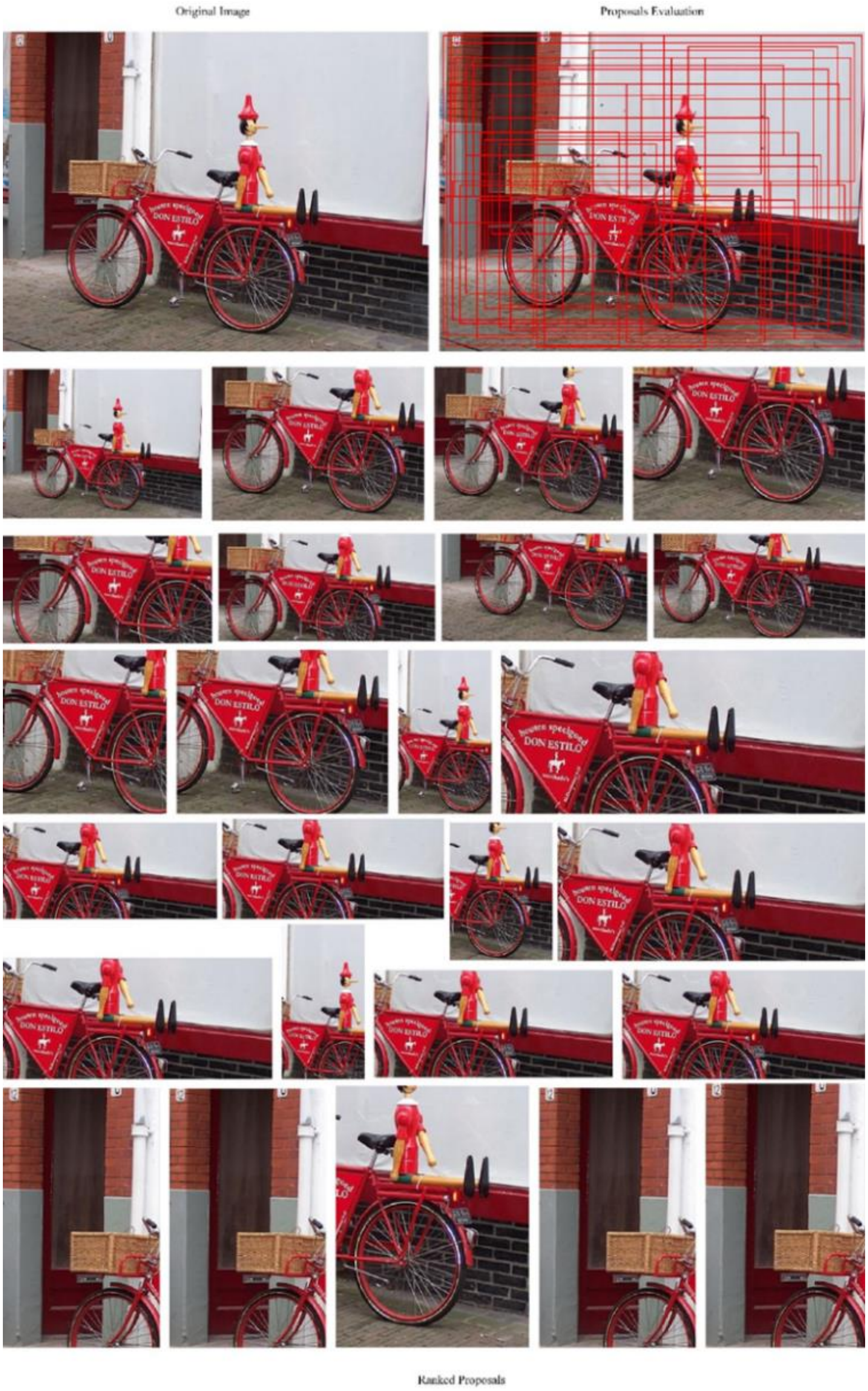

Figure 2. Obtained proposals

To obtain as many proposals as possible, the region search was diversified using a clustering technique based on different color spaces, changing initial regions, and region similarity creations. The obtained regions were grouped, and the identical ones were removed. At this point, the regions obtained after grouping are referred to as proposals (Figure 2).

The next task is to score and rank the obtained proposals. To achieve this goal, the structure edge detector was adopted to extract the edges from the original image. This detector is hailed for its relatively fast and accurate performance in edge detection. After that, the edges were connected based on their orientation similarities with adjacent edges. The eight adjacent edges, whose sum of orientation differences was above pi/2, were combined into an edge group. Further, the affinities between adjacent groups were computed, according to their mean positions and orientations. To improve computing effect, only the affinities above the threshold of 0.05 were retrained. Based on the edge groups and their affinities, the score of each proposal was calculated as follows:

For each group, a continuous value $w_b(S_i)$ was computed depending on whether a group of edges S_i is contained in the candidate bounding box b. If S_i is not fully contained in b, then $w_b(S_i)$=0. Whether S_i is fully contained in b can be judged by:

$$w_b \cdot (S_i) = 1 - \max_t \prod_j^{|T|-1} a(t_j - t_{j+1}) \tag{1}$$

where, t is the ordered path of the edge group; $|T|$ is the length of path t; a is the affinity between two edge groups in the absence of T. The path starts at $t_1 \in S_b$ and ends at $t_{|T|}$=S_i. If T does not exist, $w_b(S_i)$ equals 1.

Based on the values obtained by formula (1), the score function can be established as:

$$\frac{\sum_i w_b(S_i)\mathrm{m_i}}{2(b_w + b_h)^k} \tag{2}$$

where, b_w and b_h are the width and height of the box, respectively; k is the bias of large boxes.

Finally, the obtained proposals were ranked by the score computed by formula (2), and imported to the backbone network for proposal refinement & classification.

3.2 Proposal refinement & classification

Through the above procedure, the authors obtained a few high-quality, class-independent proposals and their scores. Yet these proposals need to be further refined to ensure the detection performance. Figure 3 provides an example of refined proposals.

Object detection systems desire for the fewest number of top-quality proposes. Hence, a proposal refinement system was employed to refine the proposals obtained in the previous stage, laying the basis for classification. In our overall design, the proposal refinement part and proposal classification part of our detector share the convolutional features to achieve robust performance.

Our system is an EfficientNet-B7 scaled up from the baseline network EfficientNet-B0, using a compound scaling mechanism. The network requires less computing cost and battery usage than other competitors. The EfficientNet [69] was adopted for its advantage over the previous networks in classification accuracy and efficiency (Table 1). Proposed by Google team in 2019, this network is a novel backbone DL architecture. The greater the scale, the better the classification accuracy. As shown in Table 1, EfficientNet-B7 achieved an accuracy of 84.3 % and 91.7% on ImageNet and CIFAR-100 datasets, respectively, with much fewer parameters than other networks.

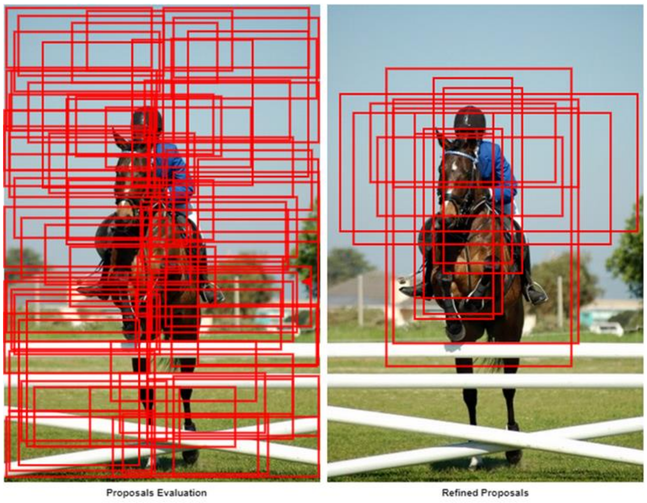

Figure 3. Refined proposals

Table 1. Top-1 accuracies of different networks

Model	Proposed year	Number of parameters	Top-1 accuracy (%)
AlexNet	2012	60 M	63.3
VGG-16	2014	138M	74.5
VGG-19	2014	143M	71.3
ResNet-50	2015	25M	77.15
Inception V3	2015	24M	78.8
Xception	2016	22.9M	79.0
InceptionResNetV2	2016	55.9M	80.3
EfficientNet-B0	2019	5.3M	76.3
EfficientNet-B7	2019	66M	84.4

The baseline network EfficientNet-B0 consists of 1 convolutional layer, 7 mobile inverted bottleneck (MBConv) blocks [70], 1 average pooling layer, and 1 fully-connected layer. MBConv is the main building block of he EfficientNet, to which squeeze-and-excitation block is added along with the Swish activation function. Each MBConv block has a different setting: The first MBConv block has a single layer with the kernel size of 3×3 and 16 output channels; the second MBConv block has two layers, each with the kernel size of 3×3 and 24 output channels; the third MBConv block has two layers, each with the kernel size of 5×5 and 40 output channels; the fourth MBConv block has three layers, each with the kernel size of 3×3 and 80 output channels; the fifth MBConv block has three layers, each with the kernel size of 3×3 and 112 output channels; the sixth MBConv block has four layers, each with the kernel of size 5×5 and 192 output channels; the seventh and the last MBConv block has a single layer with the kernel size of 3×3 and 320 output channels.

It should be noted that, to refine and classify proposals, the network after the last MBConv block was modified by adding two branches. The modified model receives the proposals generated in the first stage and the corresponding natural image. Then, the input image is passed through of the first to the fifteenth layer. To reduce computing cost and time, the proposal refinement network developed by Liu et al. [71] was adopted as a refinement branch, which is suitable for out settings. The refinement network was added behind the last MBConv block, including two refinement convolutional layers with kernel sizes of 3×3 and 5×5 , respectively. The addition reduces the number of channels from the previous layer from 320 to 128, marking the starting point of our proposal refinement.

Next, a rectified linear unit (ReLU) layer was introduced. After that, a region of interest (ROI) pooling layer was added to perform down-sampling of each initial box region, producing a feature map of the size 5×5. The down-sampling meshes the input feature map into various grids of equal width and height. Then, maximum pooling was performed on each grid. Subsequently, another fully-connected layer followed by a ReLU layer was added to output 1,024 neurons only. Further, a ranking branch composed of a fully-connected layer was arranged to recalculate the score of each proposal. This ranking branch has two output neurons, which symbolize the likelihoods of the existence of an object. Meanwhile, another branch of box regression, which is also a fully-connected layer was deployed to capture the position offsets of initial proposals, and predict the box regression values. During network training, a binary class label was also assigned to each initial proposal to check whether it is an object. The loss function can be defined as:

$$L_{obj}\left(p,u\right)=-\left[1_{\{u=1\}}\log p_1+1_{\{u\neq1\}}\log p_0\right] \tag{3}$$

where, *p* is the value computed by softmax function based on the two outputs of a fully-connected layer; u is the label of the current box. In addition, the coordinate offsets were learned by the box regression layer. The coordinates can be parameterized as:

$$\begin{aligned}
&t_x=\left(x-x_{in}\right)/w_{in},t_y=\left(y-y_{in}\right)/h_{in},\\
&t_w=\log\left(w/w_{in}\right),t_h=\log\left(h/h_{in}\right),\\
&v_x=\left(x^*-x_{in}\right)/w_{in},v_y=\left(y^*-y_{in}\right)/h_{in},\\
&v_w=\log\left(w^*/w_{in}\right),v_h=\log\left(h^*/h_{in}\right),
\end{aligned} \tag{4}$$

where, *x* and *y* are the center coordinates of a candidate box; *h* and *w* are the height and width of candidate box; *x*, x_{in}, and x^* are the predicted, input, and ground-truth abscissas of the candidate box, respectively; the parameters related to *y*, *h*, and *w* are defined similarly; *v* is the regression target; t is the predicted tuple. Thus, the loss of box regression can be described as:

$$\begin{aligned}
&L_{reg}=\sum_{i\epsilon\{x,y,w,h\}}\text{smooth}_{L_1}\left(t_i-v_i\right)\\
&smooth\$_{L_1}(x)=\begin{cases}0.5x^2 & \text{if } |x|<1\\ |x|-0.5 & \text{otherwise}\end{cases}
\end{aligned} \tag{5}$$

where, smooth $L_1(x)$ is the regression loss function. Hence, the joint loss function can be defined as:

$$L(p,u,t,v)=L_{obj}(p,u)+\lambda\cdot1_{\{u=1\}}L_{reg}(t,v) \tag{6}$$

where, λ=1 is a balance parameter.

3.3 Parameter settings

In our experiments, the proposed model was trained and tested on the trainval and test sets of PASCAL Visual Object Classes Challenge 2007 (VOC2007), respectively. The network training was performed using the Adam optimizer. From a target image, each Adam mini batch yielded 128 boxes, which were taken as training samples. The 128 training samples from each batch were equally divided into positive and negative samples: the boxes with a >0.7 overlap value with ground-truth boxes were considered positive samples, and those with an overlap value in [0.1, 0.5] were considered negative samples. The experiments lasted a total of 32 iterations. The model layers were finetuned by fixing the learning rate at 0.0001 for all 32 iterations. To train the detection module of our model, 256 object proposals were generated in each mini-batch for each image. In Fast-RCNN, the proposals with an overlap value of 0.5 with the ground-truth boxes are treated as positive samples, that is, the positive samples take up 25% of the proposals. Meanwhile, those with an overlap value in [0.1, 0.5] are considered negative samples. Furthermore, the top 1,500 proposals were selected for model training, with a fixed learning rate (0.0001) in all iterations. The model testing was carried out on the top 100 proposals per image, which is much fewer than those required by previous approaches.

4. EVALUATION AND RESULTS

The effectiveness of our model was evaluated on PSCAL VOC2007 dataset [72], the most popular benchmark in the field of object detection. The dataset contains 9,963 images with objects in 20 classes. Here, the dataset is split into a training set of 2,501 images, a validation set of 2,510 images, and a test set of 4,952 images. The images were divided into these sets along with their bounding box labels.

The overall detection performance was evaluated by metrics like mean average best overlap (MABO), detection recall (DR), and mean average precision (mAP). MABO and DR were selected to measure the positioning accuracy; mAP was chosen to assess the detection accuracy.

To validate the superiority and robustness of our method, numerous state-of-the-art methods were taken as contrastive schemes. For the comparison of positioning accuracy, the following methods were selected: Rantalankila's method [8], GOP [9], Rahtu's method [10], Objectness [11], BING [12], RandomPrim [13], Selective Search [14], CSVM [15], LPO [16], EdgeBox [17], MCG [18], Endres [19], DeepBox [20], RPN [35], DeepMask [21], and SharpMask [22]. The proposals obtained by these methods were fed into the detection module to check the quality of the proposes in the overall detection task.

For the comparison of detection accuracy (mAP) of our model on the proposed obtained by previous methods, multiple two-stage methods were chosen: R-CNN [32], SPPnet [33], Fast R-CNN [34], Faster R-CNN [35], MR-CNN [73], R-FCN [36], ION [51], HyperNet [52], OHEM [53], Craft [74], LocNet [75], R-FCN with deformable convolutional network (DCN) [76], CoupleNet [77], DeNet512 (wide) [78], FPN-Reconfig [79], DeepRegionLet [80], and DCN + R-CNN [81]. The detection efficiency of our model was compared with one-stage detectors like YOLO [39], YOLOv2 [82], SSD300 [40], deconvolutional single shot detector (DSSD) 321 [83], Deeply

Supervised Object Detector (DOSD) 300 [84], reverse connection with objectness prior network (RON) 384 [85], and CenterNet [86].

Tables 2 and 3 show the detection recalls of various methods on different number of proposals (100, 300, 500, and 1,000) at intersection over union (IoU) thresholds of 0.5 and 0.7, respectively. Figure 4 compares the detection recalls of different methods against the IoU threshold using 100 proposals per image. Figures 5 and 6 illustrate the detection recalls of various methods on different number of proposals at the IoU thresholds of 0.5 and 0.7, respectively.

Table 2. Detection recalls of various methods on different number of proposals at IoU=0.5

Methods	**Number of proposals**			
(IoU=0.5 Recall)	100	300	500	1000
Rantalankia's method	15.1	17.1	20.6	25.3
GOP	60.9	62.2	67.3	72.5
Rahtu's method	63.1	64.4	67.8	72.4
Objectness	66.2	67.5	70.6	75.2
BING	71.0	72.3	74.5	79.7
RandomPrim	71.9	73.5	75.6	80.3
Selective Search	72.4	74.9	76.2	81.6
CSVM	75.1	77.8	79.4	84.9
LPO	76.3	78.1	80.5	85.3
EdgeBox	76.4	78.7	81.2	86.9
MCG	82.9	84.5	86.5	87.5
Endres	84.5	86.2	87.9	88.1
DeepBox	85.2	87.8	88.7	92.0
RPN	90.2	90.9	91.9	92.9
DeepMaskZoom	91.4	91.8	92.3	93.4
SharpMaskZoom	91.9	92.4	92.9	93.7
Our model	92.8	93.6	93.9	94.5

Table 3. Detection recalls of various methods on different number of proposals at IoU=0.7

Methods	**Number of proposals**			
(IoU=0.7 Recall)	100	300	500	1000
Rantalankia's method	8.5	10.8	13.4	17.7
BING	25.6	27.7	30.8	34.4
CSVM	26.9	28.7	31.2	35.3
Objectness	30.3	32.5	35.5	39.6
GOP	36.7	38.3	41.1	45.2
RandomPrim	45.6	47.1	50.2	54.7
Rahtu's method	46.3	48.4	51.4	55.2
LPO	49.7	51.5	53.6	57.8
Selective Search	50.2	52.7	55.7	59.4
Endres	60.8	62.8	65.3	69.3
MCG	61.4	63.3	66.2	70.1
EdgeBox	61.7	63.9	66.8	70.9
RPN	65.5	67.7	70.8	74.9
DeepBox	71.3	73.2	76.3	80.5
DeepMaskZoom	72.1	74.3	77.2	82.5
SharpMaskZoom	75.2	77.4	80.5	84.9
Our model	77.7	79.6	82.7	86.1

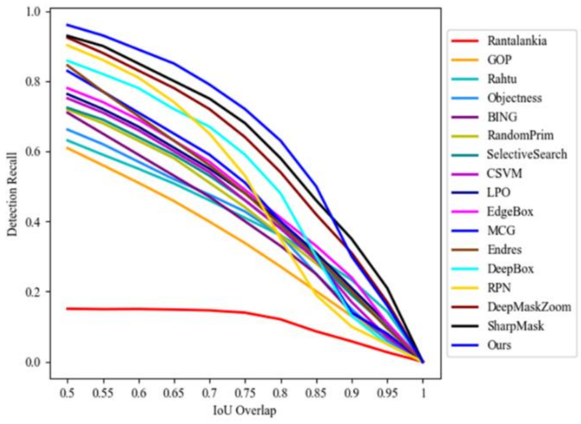

Figure 4. Detection recalls of different methods against the IoU threshold using 100 proposals per image

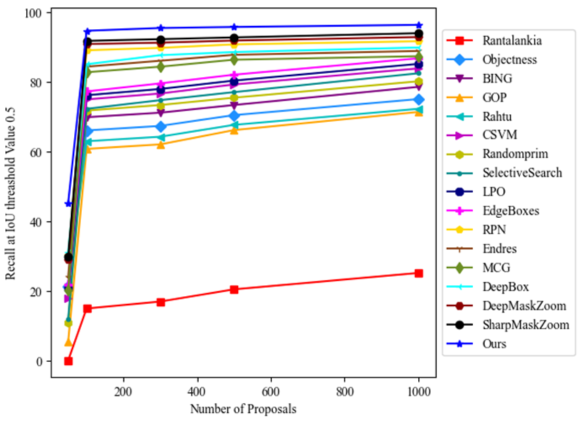

Figure 5. Detection recalls of various methods on different number of proposals at the IoU threshold of 0.5

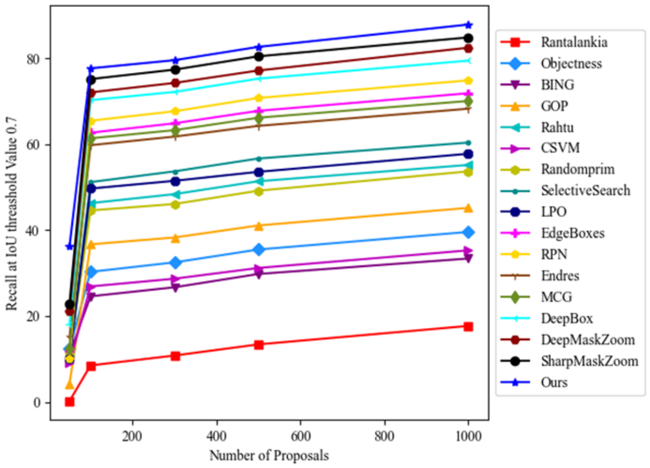

Figure 6. Detection recalls of various methods on different number of proposals at the IoU threshold of 0.7

Our model achieved higher detection recall than the previous approaches, and did well across the two IoU thresholds facing different number of proposals. The performance of our model remained robust, despite the changes in IoU threshold and the number of proposals. For 100, 300, 500, and 1,000 proposals, our model achieved recall values of 92.8%, 93.6%, 93.9%, and 94.5% at IoU=0.5, and 77.7%, 79.6%, 82.7%, and 86.1% at IoU=0.7, respectively. The good performance is the result of the sharing of convolutional features across the network.

Despite producing high-quality proposals, many state-of-the-art approaches failed to achieve a high recall, due to the limited number of candidate boxes. A few relatively competitive methods could not reach a high recall, as the proposals are loosely fitted. None of the contrastive methods achieved a robust performance at either IoU threshold.

Moreover, our model achieved a recall of 92.8% on only 100 proposals per image at IoU=0.5. The same recall was achieved by the RPN using 1,000 proposals per image. Our model could realize a high recall with a few proposals, because its proposals are highly diverse. Overall, our approach boasts robust performance on a few proposals, and acceptable performance on many proposals. On the contrary, the previous approaches were outshined by our method, whether there were many or a few proposals. Hence, this paper proposes a much more accurate method than the existing approaches, thanks to the high-quality and refinement of object proposals.

Table 4. Detection effects of different methods

Methods (MABO)	**Number of proposals**				**mAP with 100 proposals per image**
	100	300	500	1000	
Rantalankia's method	24.1	26.8	30.9	34.5	20.2
Objectness	57.2	59.7	62.1	66.2	55.2
BING	58.4	60.3	63.6	67.6	54.5
GOP	59.3	61.5	64.7	68.8	43.1
Rahtu's method	60.4	62.3	65.5	69.2	53.2
CSVM	61.3	63.4	66.6	70.9	56.6
RandomPrim	64.9	66.7	70.5	75.4	52.3
Selective Search	66.6	68.9	72.8	77.1	54.2
LPO	67.7	69.8	73.9	78.3	54.7
EdgeBox	68.5	70.2	74.7	79.9	64.1
RPN	72.3	74.2	78.9	83.9	76.7
Endres	72.8	74.9	79.6	84.1	68.6
MCG	73.2	75.8	80.2	85.5	67.2
DeepBox	73.9	76.7	80.1	85.3	70.2
DeepMaskZoom	78.4	80.8	84.3	89.4	74.4
SharpMaskZoom	79.2	81.4	85.5	90.7	75.5
Our model	81.1	82.9	87.4	92.8	78.3

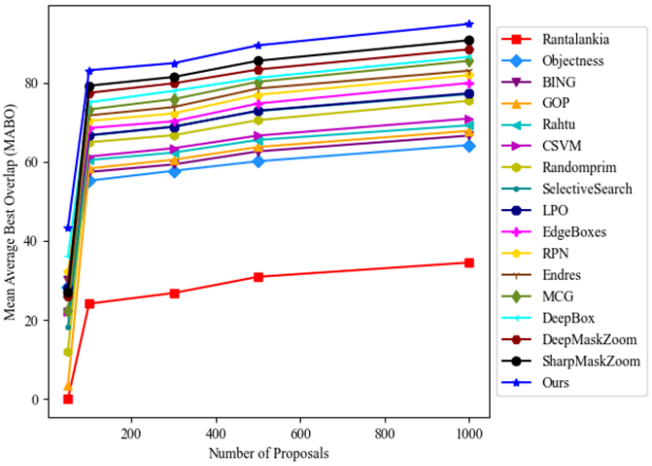

Figure 7. Positioning accuracies of different methods on different number of proposals

Table 4 compares the detection effects of different methods, and Figure 7 compares the positioning accuracies of different methods on different number of proposals. Our model achieved the highest MABOs (81.1%, 82.9%, 87.4%, and 92.8%) on 100, 300, 500, and 1,000 proposals per image, respectively. Our model had acceptable performance on a few high-quality proposals. Interestingly, the existing approaches yielded good recalls, but were inadequate to produce high MABOs. This is because of the low-quality of the numerous proposals. Only a very few approaches realized better MABO than our model. Overall, our model significantly improves the detection performance, due to the variation in object classes and low computing cost.

It can also be seen from Table 4 that, in the presence of only the top 100 proposals of each image, our model ended up with a high mAP of 78.3% by refining these high-quality proposals. Combined with the performance comparison in Figure 4, our model achieved the highest detection recall of 92.8%, when only 100 proposals were available per image. The mAP of our model was 14.2%, 23.4%, 24%, and 26% higher than that of EdgeBox, Selective Search, BING, and RandomPrim, respectively, and greater than that of any other contrastive approach. The extensive results show that our model can realize excellent recall with a few proposals, and achieve an exceptionally well defection effect.

Table 5 compares the detection accuracies of our model with other existing detectors on the Pascal VOC 2007 test set. The mAP of our model stood at 85.2%, higher than that of any other prior method. The reason is that our model generates a few high-quality and accurate proposals, and achieves a high recall. However, this ability is considered as a weakness in constative methods. In contrast to other alternatives, our model maintains the feature information, and shares the convolutional features within the network, without affecting the proposal generation performance. The feature sharing significantly improves the overall detection performance. It can also be observed that the contrastive approaches had poor real-time performance, due to their low mAPs. Therefore, the experimental results demonstrate that our model is more efficient, accurate, and useful than the contrastive methods in image object detection. Figure 8 shows the qualitative results of our model on the test set.

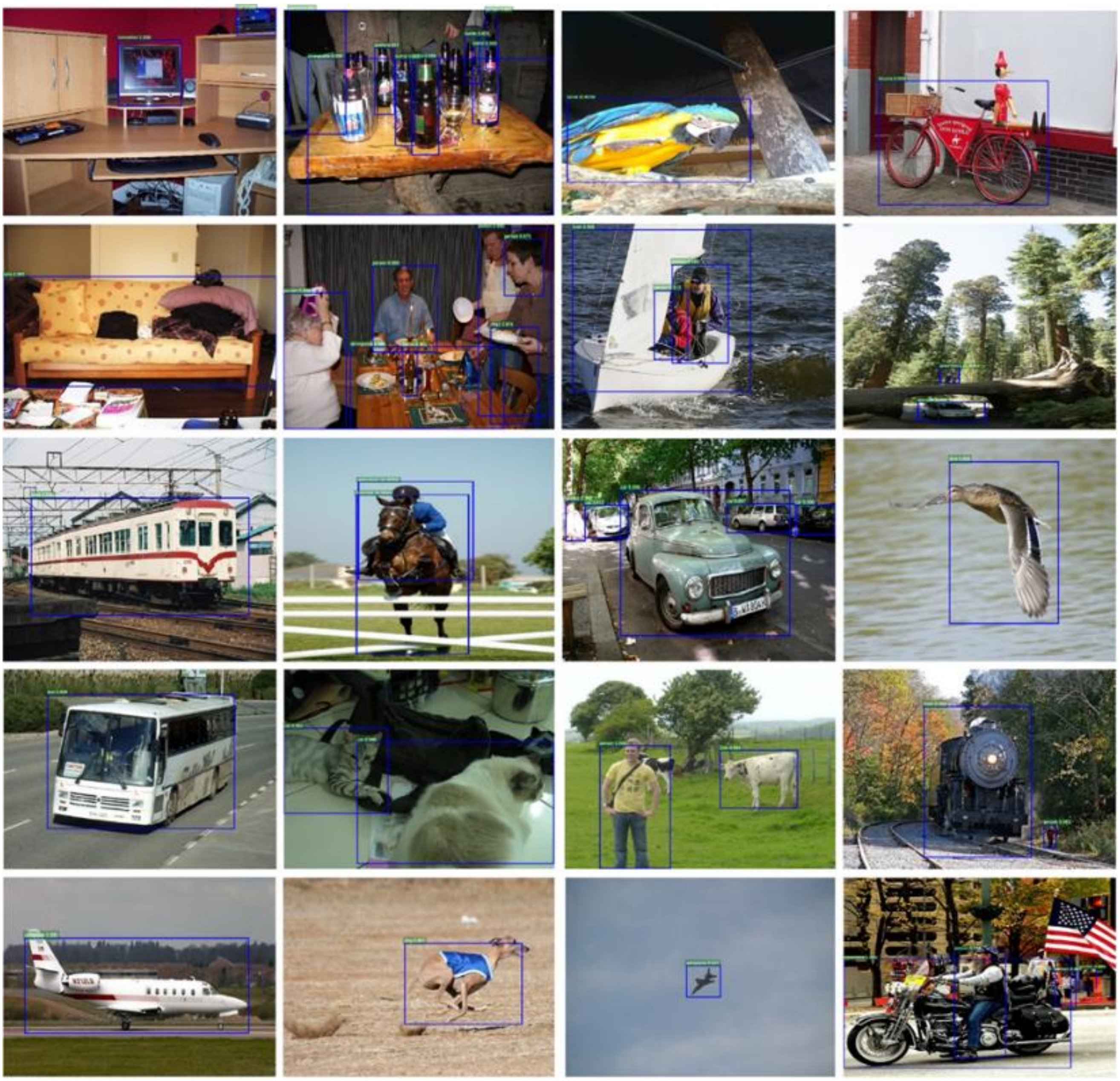

Figure 8. Qualitative results of our method

Table 5. Detection accuracies of different methods on the test set

Method	Backbone architecture	Proposed Year	Input size (Test)	mAP (%)
R-CNN [32]	AlexNet	2014	600 × 1000	50.2
R-CNN [32]	OxfordNet (VGG-16)	2014	Arbitrary	66.0
SPPnet [33]	AlexNet	2014	224 × 224	63.1
Fast R-CNN [34]	OxfordNet (VGG-16)	2014	600 × 1000	70.0
Faster R-CNN [35]	OxfordNet (VGG-16)	2015	600 × 1000	73.2

MR-CNN [73]	OxfordNet (VGG-16)	2015	Multi-Scale	78.2
R-FCN [36]	ResNet-101	2016	600 × 1000	80.5
ION [51]	OxfordNet (VGG-16)	2016	600 × 1000	76.4
HyperNet [52]	OxfordNet (VGG-16)	2016	600 × 1000	76.3
OHEM [53]	OxfordNet (VGG-16)	2016	600 × 1000	74.6
Craft [74]	OxfordNet (VGG-16)	2016	600 × 1000	75.7
LocNet [75]	OxfordNet (VGG-16)	2016	600 × 1000	78.4
R-FCN with DCN [76]	ResNet-101	2017	600 × 1000	82.6
CoupleNet [77]	ResNet-101	2017	600 × 1000	82.7
DeNet512(wide) [78]	ResNet-101	2017	512 × 512	77.1
FPN-Reconfig [79]	ResNet-101	2018	600 × 1000	82.4
DeepRegionLet [80]	ResNet-101	2018	600 × 1000	83.3
DCN + R-CNN [81]	ResNet-101 +	2018	Arbitrary	84.0
One Stage Detector				
YOLO [39]	OxfordNet (VGG-16)	2016	448 × 448	63.4
YOLOv2 [83]	DarkNet	2017	544 × 544	78.6
SSD300 [40]	OxfordNet (VGG-16)	2016	300 × 300	77.2
SSD512 [40]	OxfordNet (VGG-16)	2016	512 × 512	79.8
DSSD321 [82]	ResNet-101	2017	321 × 321	78.6
DSSD513 [82]	ResNet-101	2017	513 × 513	81.5
DOSD300 [84]	DenseNet	2017	300 _ 300	77.7
RON384 [85]	OxfordNet (VGG-16)	2017	384 × 384	75.4
CenterNet [86]	ResNet101	2019	512 × 512	78.7
Our method	EfficientNets	2021	600 × 1000	85.2

5. DISCUSSION

After reviewing on the relevant literature on object detection, this paper offers an improved method for image object detection, which realizes higher recall, MABO, and mAP than existing methods. Based on the results of comparative experiments in Section 4, this section tries to highlight the advantages of our model.

Compared with Rantalankila's method [8] and GOP [9], our model had a marked edge in recall (77.7% and 31.9%), MABO (57% and 21.8%), and mAP (58.1% and 35.2%) on 100 proposals at the IoU threshold of 0.5. A high recall and detection efficiency were realized by our model on a few or many proposals. But this is considered as a drawback in Rantalankila's method and GOP. The poor detection performance of these two methods is stemmed from the lack of direct control over proposal generation, the absence of a proposal scoring mechanism, and the generation of poor, duplicate proposals. Moreover, Rahtu's method [10] uses the first top ranking proposals instead of the top best proposals, which limits the overall detection performance. Our model is much more efficient than that method in terms of detection rate, MABO, and mAP. Our model generates a few best quality proposals, and manifests that these proposals are enough to realize good detection effect. Furthermore, our model performed better than Objectness [11] at a high IoU threshold, realizing more robust detection performance.

As for BING [12], the detection efficiency declined with the increase in IoU threshold, and some classes of objects were detected at a lower accuracy than that of our model. RandomPrim [13] neither has a scoring mechanism nor controls proposal generation. Its detection effect weakened with the growing number and random selection of proposals. This explains the huge performance gap of RandomPrim with our model. SelectiveSearch [14] involves no learning phase in proposal generation, yet still achieved fairly good performance. In contrast, our model drastically outshined SelectiveSearch by creating fewer high-quality proposals: the edges of our model in recall, MABO, and mAP were 20.4%, 14.5%, and 24.1%, respectively. Unlike SelectiveSearch, our model speeds up the detection process by scoring, ranking, and refining the proposals. Likewise, our model achieved better performance than CSVM [15] in all metrics: recall (17.7%), MABO (19.8%), and mAP (21.7%). Although the LPO [16] was better than some approaches in recall, MABO, and mAP, it was not as good as our model, for its low-quality proposals are insufficient to enhance the overall detection performance.

EdgeBox [17] could perform well on a high number of proposals, and controls proposal generation with a scoring mechanism. However, the network is not effective in proposal ranking for the purpose of robust detection. Our model surpassed EdgeBox in every evaluation metric. Similarly, our model is superior to MCG [18] in recall, MABO, and mAP for all settings. The latter's performance decreased with the growing IoU threshold. Moreover, the high-cost model proposed by Enders [19] failed to detect all the objects in every image, and was nowhere near our model in terms of the three metrics. Further, the performance DeepBox [20] was surpassed by that of our model across the board, because the network generates poorer proposals than our model, and the generated proposals could not be reduced. Our model also generated fewer proposals and reduced false positives, i.e., achieved better performance, than RPN [35], DeepMask [21], and SharpMask [22]. The advantage of our model over these three networks is small yet striking.

Once generated, the high-quality proposals can be directly applied to object detection in images. Therefore, the detection efficiency of our model was further compared with various state-of-the-art object detectors [32-36, 39-40, 51-53, 73-86]. As shown in Table 5, our model achieved the best detection accuracy, thanks to its potential to generate a few high-quality proposals. In addition to the edge in recall and proposal accuracy, our model output robust results on objects in different classes. The highest mAP of 85% shows the effectiveness of our model than prior detectors in object detection.

The key objectives of this research revolve around the generation of fewer high-quality and class-independent proposals, speeding up the detection process, and the efficacy measured by quality evaluation metrics. The contrastive methods have certain advantages and disadvantages in these aspects. But the proposed model surpassed all of them in recall, MABO, and mAP.

6. CONCLUSIONS

This paper proposes an effective model for efficient detection of image objects, adding a simple yet compelling tool to the object detector family. Our model first generates a few high-quality, class-independent, and accurate object proposals. Then, the class of each object is determined efficiently based on these proposals. In addition, convolutional features are shared across the network to maintain good recall and accuracy with a few proposals. This is considered a drawback in previous approaches, as too many proposals would hinder detection efficiency.

Through efficient proposal generation & refinement, our model can achieve a high recall on true objects, which facilitates the accurate recognition of proposals and detection of objects. Further, the efficiency of the proposed model was evaluated on the Pascal VOC 2007 test set. The experimental results show the superiority of our model over existing methods in recall, MABO, and mAP. The recall of our model at the IoU threshold of 0.5 was 92.8%, 93.6%, 93.9%, and 94.5% on 100, 300, 500, and 1,000 proposals, respectively; the MABO was 81.1%, 82.9%, 87.4%, and 92.8% on 100, 300, 500, and 1,000 proposals, respectively; the mAP was 78.3% on 100 proposals per image. Our model also outperformed the other detector on the Pascal VOC 2007 test set, with an mAP as high as 85.2%. The excellent experimental results indicate that our model is an effective and robut tool for object detection in images.

The future research will extend our model to weakly supervised object detection by generating more proposals at a higher IoU threshold and choosing as many true proposals as possible. The training process can be made more effective by mining discriminative hard negatives. Furthermore, the authors intend to learn and apply our model in diverse domains.

ACKNOWLEDGMENT

The study was supported by the "Science and Technology Research Plan Project of Hubei Provincial Department of Education, China (Grant No.: D20192902), and the study was supported by "National Statistical Science Research Project in 2020, China (Grant No.: 2020LY023)".

REFERENCES

[1] Shivappriya, S.N., Priyadarsini, M., Stateczny, A., Puttamadappa, C., Parameshachari, B.D. (2021). Cascade object detection and remote sensing object detection method based on trainable activation function. Remote Sensing, 13(2): 200. https://doi.org/10.3390/rs13020200

[2] Wang, H., Wang, Q., Li, P., Zuo, W. (2021). Multi-scale structural kernel representation for object detection. Pattern Recognition, 110: 107593. https://doi.org/10.1016/j.patcog.2020.107593

[3] Zhou, Q., Wang, J., Liu, J., Li, S., Ou, W., Jin, X. (2021). RSANet: Towards real-time object detection with residual semantic-guided attention feature pyramid network. Mobile Networks and Applications, 26(1): 77-87. https://doi.org/10.1007/s11036-020-01723-z

[4] Zhang, G., Cui, K., Wu, R., Lu, S., Tian, Y. (2021). PNPDet: Efficient few-shot detection without forgetting via plug-and-play sub-networks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3823-3832.

[5] Shin, S., Han, H., Lee, S.H. (2021). Improved YOLOv3 with duplex FPN for object detection based on deep learning. The International Journal of Electrical Engineering & Education, 0020720920983524. https://doi.org/10.1177/0020720920983524

[6] Aamir, M., Pu, Y.F., Rahman, Z., Abro, W.A., Naeem, H., Ullah, F., Badr, A.M. (2018). A hybrid proposed framework for object detection and classification. Journal of Information Processing Systems, 14(5): 1176-1194. https://doi.org/10.3745/JIPS.02.0095

[7] Aamir, M., Pu, Y.F., Rahman, Z., Tahir, M., Naeem, H., Dai, Q. (2019). A framework for automatic building detection from low-contrast satellite images. Symmetry, 11(1): 3. https://doi.org/10.3390/sym11010003

[8] Rantalankila, P., Kannala, J., Rahtu, E. (2014). Generating object segmentation proposals using global and local search. 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2417-2424. https://doi.org/10.1109/CVPR.2014.310

[9] Krähenbühl, P., Koltun, V. (2014). Geodesic object proposals. In European Conference on Computer Vision, 8693: 725–739. https://doi.org/10.1007/978-3-319-10602-1_47

[10] Rahtu, E., Kannala, J., Blaschko, M. (2011). Learning a category independent object detection cascade. In 2011 International Conference on Computer Vision, 1052-1059. https://doi.org/10.1109/ICCV.2011.6126351

[11] Alexe, B., Deselaers, T., Ferrari, V. (2012). Measuring the objectness of image windows. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(11): 2189-2202. https://doi.org/10.1109/TPAMI.2012.28

[12] Yorozu, Y., Hirano, M., Oka, K., Tagawa, Y. (2014). Binarized normed gradients for objectness estimation. Computer Visiion and Pattern Recognition, 3286-3293.

[13] Manen, S., Guillaumin, M., Van Gool, L. (2013). Prime object proposals with randomized prim's algorithm. In Proceedings of the IEEE International Conference on Computer Vision, pp. 2536–2543.

[14] Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W. (2013). Selective search for object recognition. International Journal of Computer Vision, 104(2): 154-171. https://doi.org/10.1007/s11263-013-0620-5

[15] Zhang, Z., Torr, P.H. (2015). Object proposal generation using two-stage cascade SVMs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(1): 102-115. https://doi.org/10.1109/TPAMI.2015.2430348

[16] Krahenbuhl, P., Koltun, V. (2015). Learning to propose objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1574–1582.

[17] Zitnick, C.L., Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In European Conference on Computer Vision, pp. 391-405. https://doi.org/10.1007/978-3-319-10602-1_26

[18] Arbeláez, P., Pont-Tuset, J., Barron, J.T., Marques, F., Malik, J. (2014). Multiscale combinatorial grouping. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 328–335.

[19] Endres, I., Hoiem, D. (2013). Category-independent object proposals with diverse ranking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(2): 222-234. https://doi.org/10.1109/TPAMI.2013.122

[20] Kuo, W., Hariharan, B., Malik, J. (2015). Deepbox: Learning objectness with convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, pp. 2479–2487.

[21] Pinheiro, P.O., Collobert, R., Dollár, P. (2015). Learning to segment object candidates. arXiv preprint arXiv:1506.06204.

[22] Pinheiro, P.O., Lin, T.Y., Collobert, R., Dollár, P. (2016). Learning to refine object segments. In European Conference on Computer Vision, pp. 75-91. https://doi.org/10.1007/978-3-319-46448-0_5

[23] Carreira, J., Sminchisescu, C. (2011). CPMC: Automatic object segmentation using constrained parametric min-cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(7): 1312-1328. https://doi.org/10.1109/TPAMI.2011.231

[24] Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25: 1097-1105.

[25] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778.

[26] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700-4708.

[27] Lin, M., Chen, Q., Yan, S. (2013). Network in network. CoRR, abs/1312.4400, 2013. http://arxiv.org/abs/1312.4400

[28] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9.

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

[30] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1): 1929-1958.

[31] Ioffe, S., Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning, 37: 448-456.

[32] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587.

[33] He, K., Zhang, X., Ren, S., Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9): 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824

[34] Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, pp. 1440-1448.

[35] Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497.

[36] Dai, J., Li, Y., He, K., Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. arXiv preprint arXiv:1605.06409.

[37] He, K., Gkioxari, G., Dollár, P., Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, pp. 2961-2969.

[38] Cai, Z., Vasconcelos, N. (2018). Cascade R-CNN: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6154–6162.

[39] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788.

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

[41] Prati, A., Shan, C., Wang, K.I.K. (2019). Sensors, vision and networks: From video surveillance to activity recognition and health monitoring. Journal of Ambient Intelligence and Smart Environments, 11(1): 5-22. https://doi.org/10.3233/AIS-180510

[42] Liu, R., Miao, Q., Song, J., Quan, Y., Li, Y., Xu, P., Dai, J. (2019). Multiscale road centerlines extraction from high-resolution aerial imagery. Neurocomputing, 329, 384-396. https://doi.org/10.1016/j.neucom.2018.10.036

[43] Aamir, M., Pu, Y.F., Rahman, Z., Tahir, M., Naeem, H., Dai, Q. (2019). A framework for automatic building detection from low-contrast satellite images. Symmetry, 11(1): 3. https://doi.org/10.3390/sym11010003

[44] Van Gansbeke, W., De Brabandere, B., Neven, D., Proesmans, M., Van Gool, L. (2019). End-to-end lane detection through differentiable least-squares fitting. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.

[45] Ranjan, R., Patel, V.M., Chellappa, R. (2017). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(1): 121-135. https://doi.org/10.1109/TPAMI.2017.2781233

[46] Milton, M.A.A. (2019). Towards pedestrian detection using Retinanet in ECCV 2018 wider pedestrian detection challenge. arXiv preprint arXiv:1902.01031.

[47] Kaya, E.C., Alatan, A.A. (2018). Improving Proposal-Based Object Detection Using Convolutional Context Features. In 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 1308-1312. https://doi.org/10.1109/ICIP.2018.8451686.

[48] Liu, B., Zhao, W., Sun, Q. (2017). Study of object detection based on Faster R-CNN. In 2017 Chinese Automation Congress (CAC), pp. 6233-6236. https://doi.org/10.1109/CAC.2017.8243900.

[49] Zhang, Y., Chen, Y., Huang, C., Gao, M. (2019). Object detection network based on feature fusion and attention mechanism. Future Internet, 11(1): 9. https://doi.org/10.3390/fi11010009

[50] Xiao, Y., Wang, X., Zhang, P., Meng, F., Shao, F. (2020). Object Detection Based on Faster R-CNN Algorithm with Skip Pooling and Fusion of Contextual Information. Sensors, 20(19): 5490. https://doi.org/10.3390/s20195490

[51] Bell, S., Zitnick, C.L., Bala, K., Girshick, R. (2016). Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2874–2883.

[52] Kong, T., Yao, A., Chen, Y., Sun, F. (2016). Hypernet: Towards accurate region proposal generation and joint object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 845–853.

[53] Shrivastava, A., Gupta, A., Girshick, R. (2016). Training region-based object detectors with online hard example mining. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 761-769.

[54] Zhao, H., Zhou, Y., Zhang, L., Peng, Y., Hu, X., Peng, H., Cai, X. (2020). Mixed YOLOv3-LITE: A lightweight real-time object detection method. Sensors, 20(7): 1861. https://doi.org/10.3390/s20071861

[55] Jiang, D., Sun, B., Su, S., Zuo, Z., Wu, P., Tan, X. (2020). FASSD: A feature fusion and spatial attention-based single shot detector for small object detection. Electronics, 9(9): 1536. https://doi.org/10.3390/electronics9091536

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

[57] Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C. (2017). Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659.

[58] Li, Z., Zhou, F. (2017). FSSD: feature fusion single shot multibox detector. arXiv preprint arXiv:1712.00960.

[59] Jeong, J., Park, H., Kwak, N. (2017). Enhancement of SSD by concatenating feature maps for object detection. arXiv preprint arXiv:1705.09587.

[60] Leng, J., Liu, Y. (2019). An enhanced SSD with feature fusion and visual reasoning for object detection. Neural Computing and Applications, 31(10): 6549-6558. https://doi.org/10.1007/s00521-018-3486-1

[61] Wang, J., Chen, K., Yang, S., Loy, C.C., Lin, D. (2019). Region proposal by guided anchoring. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2960–2969.

[62] Wan, F., Wei, P., Jiao, J., Han, Z., Ye, Q. (2018). Min-entropy latent model for weakly supervised object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1297-1306.

[63] Arun, A., Jawahar, C.V., Kumar, M.P. (2019). Dissimilarity coefficient based weakly supervised object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9432-9441.

[64] Wan, F., Liu, C., Ke, W., Ji, X., Jiao, J., Ye, Q. (2019). C-mil: Continuation multiple instance learning for weakly supervised object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2199–2208.

[65] Shen, Y., Ji, R., Wang, Y., Wu, Y., Cao, L. (2019). Cyclic guidance for weakly supervised joint detection and segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 697–707.

[66] Yang, K., Li, D., Dou, Y. (2019). Towards precise end-to-end weakly supervised object detection network. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8372–8381.

[67] Tang, P., Wang, X., Bai, S., Shen, W., Bai, X., Liu, W., Yuille, A. (2018). PCL: Proposal cluster learning for weakly supervised object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(1): 176-191. https://doi.org/10.1109/TPAMI.2018.2876304

[68] Felzenszwalb, P.F., Huttenlocher, D.P. (2004). Efficient graph-based image segmentation. International Journal of Computer Vision, 59(2): 167-181. https://doi.org/10.1023/B:VISI.0000022288.19776.77

[69] Tan, M., Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning, 97: 6105-6114.

[70] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510-4520.

[71] Liu, Y., Li, S., Cheng, M.M. (2020). Refinedbox: Refining for fewer and high-quality object proposals. Neurocomputing, 406: 106-116. https://doi.org/10.1016/j.neucom.2020.04.017

[72] Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A. (2007). The PASCAL visual object classes challenge 2007 (VOC2007) results.

[73] Gidaris, S., Komodakis, N. (2015). Object detection via a multi-region and semantic segmentation-aware CNN model. In Proceedings of the IEEE International Conference on Computer Vision, pp. 1134-1142.

[74] Yang, B., Yan, J., Lei, Z., Li, S.Z. (2016). Craft objects from images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6043-6051.

[75] Gidaris, S., Komodakis, N. (2016). Locnet: Improving localization accuracy for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 789-798.

[76] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y. (2017). Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, pp. 764-773.

[77] Zhu, Y., Zhao, C., Wang, J., Zhao, X., Wu, Y., Lu, H. (2017). Couplenet: Coupling global structure with local parts for object detection. In Proceedings of the IEEE International Conference on Computer Vision, pp. 4126-4134.

[78] Tychsen-Smith, L., Petersson, L. (2017). Denet: Scalable real-time object detection with directed sparse sampling. In Proceedings of the IEEE International Conference on Computer Vision, pp. 428-436.

[79] Kong, T., Sun, F., Tan, C., Liu, H., Huang, W. (2018). Deep feature pyramid reconfiguration for object detection. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 169-185.

[80] Xu, H., Lv, X., Wang, X., Ren, Z., Bodla, N., Chellappa,

R. (2018). Deep regionlets for object detection. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 798-814.

[81] Cheng, B., Wei, Y., Shi, H., Feris, R., Xiong, J., Huang, T. (2018). Revisiting RCNN: On awakening the classification power of faster RCNN. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 453-468.

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

[83] Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C. (2017). Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659.

[84] Shen, Z., Liu, Z., Li, J., Jiang, Y.G., Chen, Y., Xue, X. (2017). DSOD: Learning deeply supervised object detectors from scratch. In Proceedings of the IEEE International Conference on Computer Vision, pp. 1919–1927.

[85] Kong, T., Sun, F., Yao, A., Liu, H., Lu, M., Chen, Y. (2017). Ron: Reverse connection with objectness prior networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5936-5944.

[86] Zhou, X., Wang, D., Krähenbühl, P. (2019). Objects as points. arXiv preprint arXiv:1904.07850.