IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 5, 2021, accepted March 21, 2021, date of publication March 24, 2021, date of current version April 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3068486

Single Shot Multibox Detector With Deconvolutional Region Magnification Procedure

YIZHONG WANG[1], PENGHUI NIU[1], XIAOYONG GUO[1], GUOWEI YANG[1], AND JUN CHEN[2], (Senior Member, IEEE)
[1]College of Electronic Information and Automation, Tianjin University of Science and Technology, Tianjin 300222, China
[2]Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada

Corresponding author: Xiaoyong Guo (gxyauthor@tust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 11604240 and Grant 51805370, and in part by the Industry-University Cooperation and Education Project of the Ministry of Education of China under Grant 201801003010.

ABSTRACT In this paper, we make an effort to improve the accuracy of small and medium object detections of SSD (Single Shot Multibox Detector). To this end, we introduce a deconvolutional region magnification procedure in which the existing layers in SSD play a role in the region proposal network and the proposed regions are magnified for recognition. Moreover, features are also extracted from a shallow layer and a new feature pyramid is constructed on top of these structures. Then, features are contacted and fed into classification and regression modules as in SSD. The weights of the present model are obtained via a pre-training-re-training strategy. By evaluating the model performance on a test set assembled by the samples in the PASCAL VOC and MS COCO datasets, the present model shows that the mAPs (mean average precisions) of small and medium object detections are 42.4% and 74.7% respectively, which are 27.1% and 15.6% better than SSD. This proves the effectiveness of our proposed method.

INDEX TERMS Deep learning, object detection, SSD, deconvolution.

I. INTRODUCTION

Recognition of objects and regions in images along with their location and classification is the central topic of computer vision that has attracted enormous attention for decades. Recently, significant improvement for object detection arises due to the emergence of the deep learning techniques [1], [2], which is a powerful method for learning feature representations automatically from raw input data. Basing on the deep convolutional neural networks (CNNs), there is an increasing number of models and applications devoted to design the object detection system, the so-called object detector. The Overfeat Network [3] is the first Deep Learning object detector which employs CNNs after a sliding window segmentation. It segments each image into several parts and does classification on each part using an individual CNN. Subsequently, the final location and classification predictions are generated by combining outputs of the previous two processes. The highly influential successors that are designed basing on the pipeline of this two steps idea include the Region Convolutional Network (R-CNN) [4], the Fast-RCNN [5], the Faster-RCNN [6], as well as the extended faster-RCNN with a position-sensitive ROI (region of interest) pooling [7]. Although these models have achieved better accuracy, there are also certain models in which the predictions of class probabilities and object locations are combined into a single step. The single-step models have the advantage of the real-time speed and memory saving while maintaining competitive accuracy. The most popular single step object detectors on the market are the Single Shot Multibox Detector (SSD) [8] and the You Only Look Once (YOLO) [9], [10]. The former is the first model to propose training on a feature pyramid in which default boxes are generated from each grid cell on each feature maps, and the later constructed in the same vein as the former but with only one feature map for classification and generated two default boxes for each grid cell directly cropped on the input image. On the PASCAL VOC2007 test, SSD can achieve 74.3% mAP (mean average precisions) at 59 FPS (frames per second) on an Nvidia Titan X for 300 × 300 input, outperforming state-of-the-art methods [8]. Since there is only one early layer assigned to collect low-lying features, the semantic information is not enough resulting in poor performance on small and medium object detections. A solution to this annoying issue is highly

The associate editor coordinating the review of this manuscript and approving it for publication was Kok-Lim Alvin Yau.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

desirable. Nowadays, there are many improvements with many refinements. For example, DSSD (Deconvolutional SSD) in which the backbone network VGG16 is replaced by a more powerful one (i.e., ResNet-101) and followed by an hourglass network structure [11], RSSD (Rainbow SSD) with a rainbow concatenation module [12], DSOD (Deeply Supervised Object Detector) which is trained from scratch and designs a DenseNet architecture to improve the parameter efficiency [13], and FSSD (Feature Fusion SSD) with an elaborately designed feature fusion module [14] are proposed successively. For extensive reviews, see [15], [16].

Inspired by these advances, in this paper, we focus on improving the performance of the small and medium object detections of SSD. To achieve this, we propose a new method in a way that combines the essence of region proposal network (RPN) in faster-RCNN and deconvolution operation in DSSD. To enhance small and medium object detections, low-level features are extracted via a deconvolutional region magnification procedure in which predicted boxes of SSD layers are regarded as region proposals and a deconvolutional layer is introduced to magnification. Our intuition is that larger feature maps can lead to higher classification accuracy. On the other hand, the feature extraction is reinforced with an existing convolutional layer in the backbone network VGG16. This layer is superficial and more information about small and medium objects may be preserved. Features that are obtained from newly added layers are contacted with features from a newly created feature pyramid which is inspired by the feature pyramid network (FPN) [17] and fed to classification and regression. By a training strategy involving a pre-train stage followed by a re-train stage, we obtain the weights of the present model. It is shown that our strategy is an effective way that makes mAPs of small and medium objects on the test nearly 27.1% and 15.6% higher than SSD.

The paper is organized as follows. In the subsequent section, we begin by reviewing the structure of SSD and then describe the newly added layers with the underlying motivation. In section 3, training and testing are made, and the outputs are provided. The performance of the present model is compared not only with SSD but also with other state-of-the-art SSD-based models that are designed for object detection across different scales on both accuracy and speed. The conclusion is given in Sec. 4.

II. MODEL

In this section, the architecture and salient properties of SSD are briefly outlined. Then the main ingredients of our design strategies are supplied and explained in detail.

A. SSD

In SSD, the idea of anchor boxes such as these in RPN and multiscale features maps such as in the FPN are combined to achieve a fast detection speed while still retaining a high detection quality. The sketch of SSD architecture is shown in Fig. 1, which includes a feature pyramid placed on top of a backbone convolutional base (e.g., VGG16) and is followed by non-maximum suppression (NMS) to produce the final detection. In the feature pyramid architecture, each layer plays a specific role to detect objects in different scales.

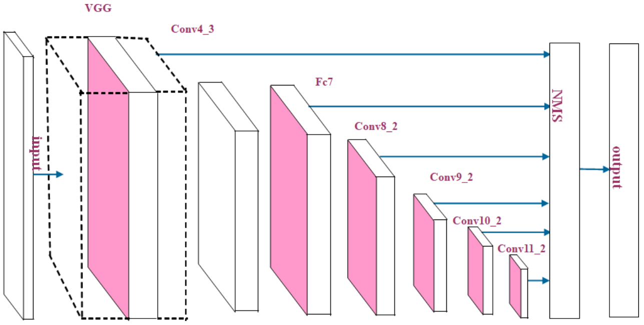


FIGURE 1. **(Color online) Sketch of SSD. A feature pyramid consisting by six convolutional layers is placed after a backbone network (VGG16). Features extracted from this pyramid are fed into classification and regression modules, and a non-maximum suppression (NMS) is also applied.**

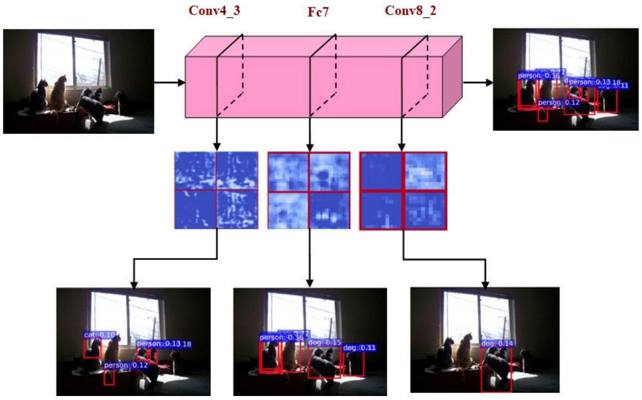


FIGURE 2. **(Color online) Visualization of feature maps and detection results corresponding to layers *conv*4_3, *fc*7, and *conv*8_2 in SSD.**

SSD generates the detection results directly from feature maps in different levels. The low-lying feature maps may contain essential location information, but the semantic and context information may be insufficient. Besides, small objects may lose their information when passes through the backbone network resulting in some missing detections. In SSD, there is only one low-level layer that is allotted to detect small objects, i.e. *conv*4_3 in the existing VGG16. The detailed information may not enough for correctly recognizing small objects. However, there are five layers above *conv*4_3 with decreasing size and resolution. These layers can give enough information for large object detection but still insufficient for medium object detection. For example, in Fig. 2, we visualize feature maps with detection results from layers *conv*4_3, *fc*7, and *conv*8_2 in SSD. It is shown that the confidence for category cat is ranging from 0.1 to 0.36, and in certain cases, the cat is incorrectly attributed as a person or dog. The detection results from low-lying feature map *conv*4_3 are even worst, where only one cat is correctly recognized with a small confidence 0.1. If we set a confidence threshold higher than that value, such as 0.3, there will be no recognition at all. For other feature maps, the confidences are increased but the

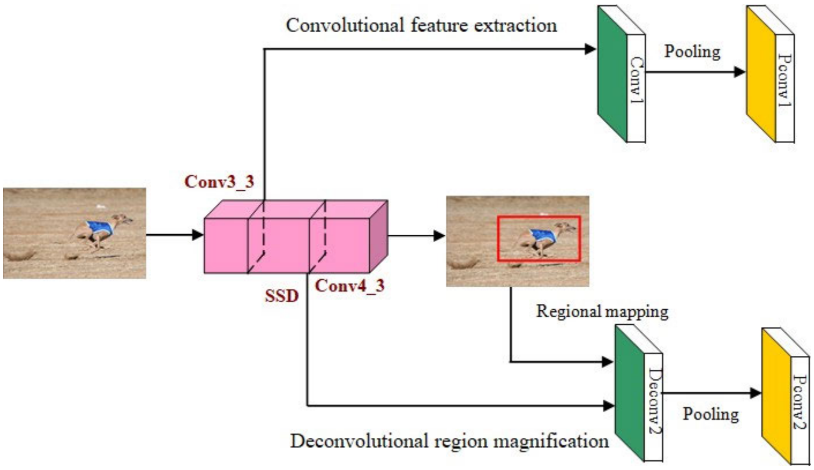


FIGURE 3. (Color online) Improvements for small and medium object detections. Here, an enlarged feature map is obtained from the deconvolution layer *Deconv*2 that follows the convolutional layer *conv*4_3, and features are also extracted from an existing layer *conv*3_3 which is shallow than *conv*4_3.

accuracies are still worried. As a result, there is a large room for improving the small and medium object detection under the framework of SSD.

B. IMPROVEMENTS FOR SMALL AND MEDIUM OBDECT DETECTIONS

The strategy to improve the detection of the small and medium objects is twofold as sketched in Fig. 3. Additional features are extracted from an existing layer *conv*3_3 in the backbone network. The intuition is that small and medium objects may not even have any information at the very top layers. A shallow layer may reserve more information about these objects. If the chosen layer is too shallow, it will be not enough semantic information. Thus, we choose a lower layer next to *conv*4_3 to reconcile this trade-off. Moreover, the deconvolutional region magnification procedure on feature maps extracted by the layer *conv*4_3 is made. The deconvolutional region magnification procedure includes a series of operations and we elaborate in the following. The deconvolution operation is taken to increase the resolution of the feature maps. After the deconvolution operation, the size of an output feature map d is increased as

$$d = s \times (i - 1) + k - 2p \tag{1}$$

where s is the number of strides; k is the size of the deconvolution filter; i is the size of the input feature map; p is the number of zero padding. Since the original size of feature maps corresponding to *conv*4_3 are 38 × 38, we introduce a deconvolutional layer *Deconv2* with $s = 8$, $k = 6$, and $p = 1$ giving rise to output feature maps of size 300 × 300. This size is equal to the input image and features of small and medium objects may be easily captured. The size of the low-lying feature map is large, which may contain essential location information, but the semantic and context information may be insufficient. Therefore, the small objects are mainly detected by the low-lying feature map, and the medium objects are detected by the high-lying feature map. Same as the RPN in Faster-RCNN, we introduce the concept of region proposal which is just predicted boxes by SSD architecture remained in our model. Then, the region proposals are mapped to the enlarged feature maps according to the following formulae

$$r_w = d_w \times \frac{f_w}{img_w}, \qquad r_h = d_h \times \frac{f_h}{img_h} \tag{2}$$

where $r_{w/h}$ is the width/height of the region proposal on the enlarged feature map; $d_{w/h}$ is the width/height of the region proposal on the input feature map; $f_{w/h}$ is the width/height of the input feature map; $img_{w/h}$ denotes the size of the input image. The proposed regions are cropped from the enlarged feature map when all region proposals are mapped. A maximum pooling is applied to resize each proposed region to 38 × 38. Feature maps in other channels are all processed by the same procedure. Although the final output of the deconvolutional region magnification procedure has the same size as the original *conv*4_3, it may contain more information of the target object and makes the detection of small and medium objects easier.

An example is shown in Fig. 4. After the backbone network, an input image of size 300 × 300 turns out to be a set of feature maps. We visualize one of them that associate with *conv*4_3 as shown in Fig. 4(b). The deconvolutional layer *Deconv*2 deconvolutes this feature map to the size 300 × 300. After cropping and maximum pooling, the proposed region returns to the size of 38 × 38. Notice that the activated region in feature maps Fig. 4(b) and Fig. 4(d) have the same shape, but feature map Fig. 4(d) contains more information that makes the detection easy. In fact, the deconvolutional region magnification procedure can be regarded as a "zoom in"

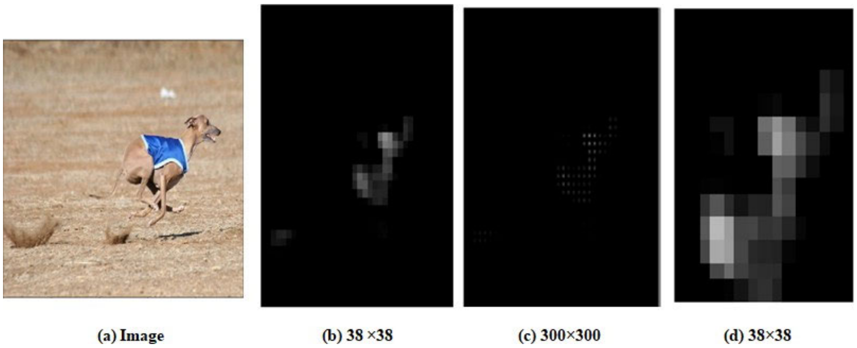


FIGURE 4. (Color online) Visualization of feature maps in the deconvolutional region magnification procedure. Here, (a) is the input image of resolution 300 × 300, (b) is the feature map corresponding to convolutional layer conv4 3, (c) is the deconvoluted feature map, and (d) is the final output of the procedure.

operation. In the detection pipeline, feature maps corresponding to *conv*4_3 are all replaced by those zoomed feature maps.

C. ARCHITECTURE

The higher-level layers may contain more semantic information and correspond to larger receptive field. Therefore, it is responsible for the detection of large objects. However, in the present model layers in SSD only plays a role of RPN such that we need to design a new feature pyramid. To this end, we gather all feature maps from existing layers *Fc*7, *conv*8_2, *conv*9_2, *conv*10_2, and *conv*11_2 assembling a new feature pyramid. To distinguish with SSD, layers in this new feature pyramid are labeled as *conv*3, *conv*4, *conv*5, *conv*6, and *conv*7. As shown in Fig. 5, these newly added layers not only have the same hyper-parameters such as kernel size of filters but also share the same weight and feature map as their counterparts in existing SSD. For each grid on each feature map, default boxes are generated in the same way as in SSD, and scores for each category and offsets for bounding boxes are predicted.

The feature maps corresponding to the deconvolutional layer *Deconv*2 should also be assigned with default boxes. Since these feature maps are cropped from an enlarged one, the mapping between the default box on a feature map and the bounding box on an input image should be modified. This mapping in SSD is expressed as

$$\begin{aligned} img_{cx} &= \frac{c_x}{f_w} img_w, img_{cy} = \frac{c_y}{f_h} img_h, \\ x_{min} &= img_{cx} - \frac{w_k}{2}, x_{max} = img_{cx} + \frac{w_k}{2}, \\ y_{min} &= img_{cy} - \frac{h_k}{2}, y_{max} = img_{cy} + \frac{h_k}{2} \end{aligned} \tag{3}$$

where $c_{x/y}$ is the center coordinate of the default box on the feature map; $img_{cx/cy}$ is the center coordinate of the bounding box; w_k/h_k is the width/height of the bounding box with $(x_{min}, y_{min}, x_{max}, y_{max})$ being its top left and bottom right coordinates. For a feature map after the deconvolutional region magnification procedure, Eq. 3 should be modified as

$$f_x^{center} = c_x \times \frac{\bar{x}_{max} - \bar{x}_{min}}{f_w} + \bar{x}_{min},$$

$$f_y^{center} = c_y \times \frac{\bar{y}_{max} - \bar{y}_{min}}{f_h} + \bar{y}_{min} \tag{4}$$

where $f_{x/y}^{center}$ is the center coordinate of the bounding box with $(\bar{x}_{min}, \bar{y}_{min}, \bar{x}_{max}, \bar{y}_{max})$ being its top left and bottom right coordinates. Since additional layers are stacked on the building block of SSD, we add seven convolutional layers for classification and other seven convolutional layers for bounding box regression. In each of these layers, we use 3×3 filter with $L_n = k \times 4$ and $C_n = k \times c$ channels for location and classification predictions, where c is the number of classes and k is the number of default boxes on each grid. The default boxes predicted scores and offsets for bounding boxes for each layer are contacted and fed into a combined loss function in the same way as SSD. With those improvements, we plot the overall architecture of our model in Fig. 6. Here, layers in SSD are depicted in the blue box, layers that improve small and medium object detections are depicted in green boxes, the new feature pyramid is depicted in the red box, and classification and regression modules are depicted in the yellow box.

D. LOSS FUNCTION

In this paper, a multi-task loss function is used [8]. The overall loss function is the weighted sum of the confidence loss (*conf*) and the localization loss (*loc*):

$$L(x, c, l, g) = \frac{1}{N}\left(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)\right) \tag{5}$$

where N is the number of matched default boxes, if $N = 0$, we set the loss to 0; α is the weight term of localization loss which is set to 1.

The confidence loss is the softmax loss over multiple classes confidences (c).

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} \log\left(\hat{c}_i^p\right) - \sum_{i \in Neg} \log\left(\hat{c}_i^o\right) \tag{6}$$

where $\hat{c}_i^p = \exp(c_i^p) / \sum_p \exp(c_i^p)$; $x_{ij}^p = \{1, 0\}$ is an indicator for matching the i-th default box to the j-th ground truth box

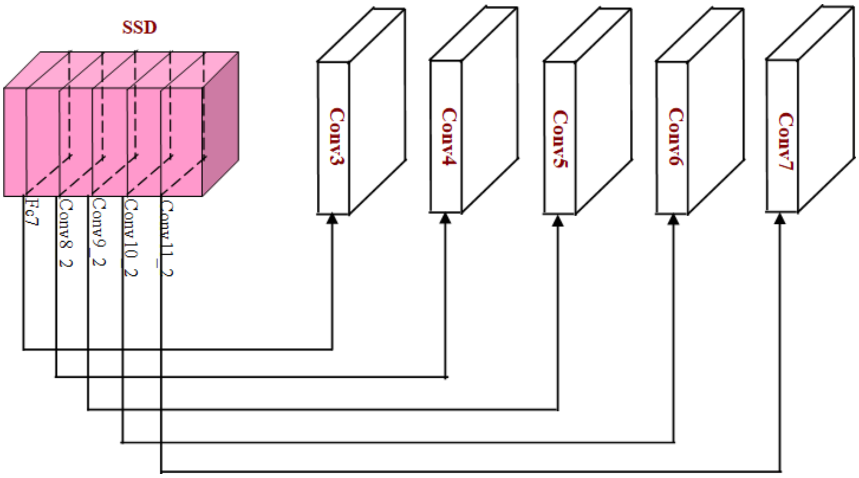


FIGURE 5. (Color online) A new feature pyramid on top of existing SSD. Here, new layers are labeled as *conv*3, *conv*4, *conv*5, *conv*6, and *conv*7. Each of these layer plays the same role as its counterpart in SSD.

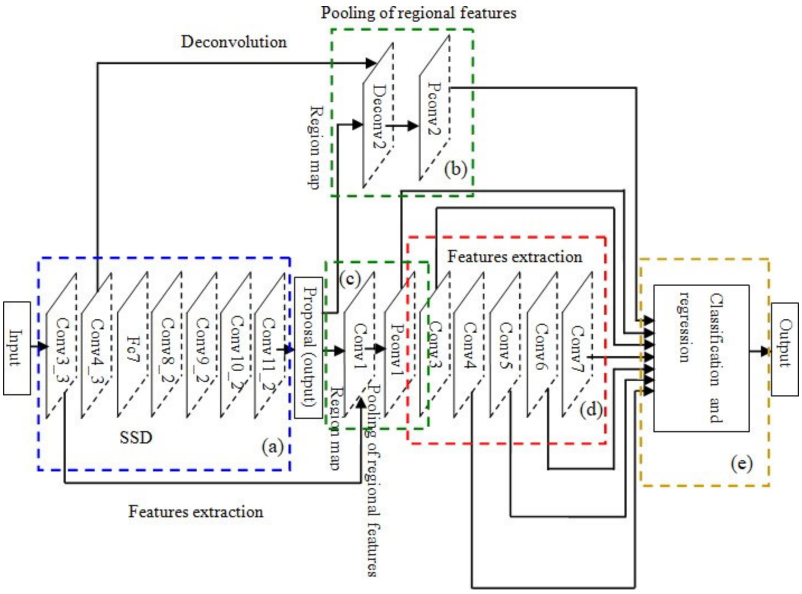


FIGURE 6. (Color online) Sketch of present model. Here, we plot every component in specific dashed boxes: (a) layers in SSD; (b) and (c) layers that improve small and medium object detections; (d) the new feature pyramid; (e) classification and regression modules.

(g) of category p. c_i^p is the output value corresponding to the i-th predicted box (l) of category p.

The localization loss is a Smooth $L1$ loss.

$$L_{loc} = (x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{ij}^k \, \mathrm{smooth}_{L1} \left(l_i^m - \hat{g}_j^m \right)$$
$$\hat{g}_j^{cx} = \left(g_j^{cx} - d_i^{cx} \right) / d_i^w, \hat{g}_j^{cy} = \left(g_j^{cy} - d_i^{cy} \right) / d_i^h$$
$$\hat{g}_j^w = \log \left(\frac{g_j^w}{d_i^w} \right), \hat{g}_j^h = \log \left(\frac{g_j^h}{d_i^h} \right) \tag{7}$$

where Pos is the set of positive samples; $\hat{g}_j^m$ is the coordinates of the corrected ground truth box; N is the number of matched default boxes, if $N = 0$, we set the loss to 0. (cx, cy) is the center of the default bounding box (d); w and h are the width and height of the default box.

III. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we are in a position to perform the training and examine the performance of the present model.

A. EXPERIMENTAL SETTING AND TRAINING STRATEGY

The code is built on Caffe [18]. We train the present model on a computer with ubuntu16.04, Intel Xeon E5-2640 v4 CPU, and eight Nvidia Titan Xp GPUs with graphic memory of 12GB. To verify the efficiency of our model for the

TABLE 1. Ablation experiment of medium object detection.

	SSD	Our model			
Deconvolution		√	√	√	√
Region proposal			√		√
Feature pyramid				√	√
mAP	59.1%	65.7%	67.3%	69.5%	74.7%

TABLE 2. Ablation experiment of small object detection.

	SSD	Our model			
Deconvolution		√	√	√	√
Region proposal			√		√
Feature pyramid				√	√
mAP	15.3%	29.7%	34.2%	36.1%	42.4%

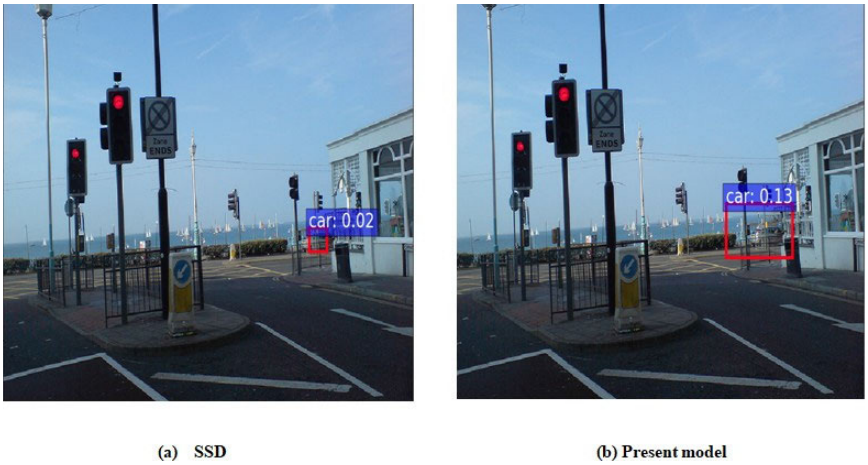
experiments, the MS COCO evaluation metrics [19] are adopted, which divide the objects into three scales according to their areas: small (area $< 32^2$), medium ($32^2 <$ area $< 96^2$), large (area $> 96^2$). According to these evaluation metrics, we select seven classes (i.e., bicycle, bus, car, cat, dog, motorbike, and person) from PASCAL VOC and MS COCO datasets, which all meet the definition of the small and medium objects. Moreover, the number of small and medium objects within these classes is larger than other classes. By these pictures, we assemble a pre-training dataset on which layers within SSD is pre-trained using the same training policy as in Literature [8]. In the following, the pre-trained SSD is also called SSD although its weights are different from that in Ref. [8]. Then, we pick up 3376 pictures from PASCAL VOC and MS COCO assembling a re-training dataset. The present model with weights from pre-training is re-trained on this dataset. The parameters for all the newly added convolutional layers are initialized with the xavier method [20]. During the pre-training, we minimize the joint localization and confidence loss. We apply the same matching strategy, hard negative mining strategy, and data augmentation as described in Ref. [8]. Using the SGD (stochastic gradient descent) with initial learning rate 10^{-4}, 0.9 momentum, 0.0005 weight decay, and batch size 20, the optimization is done after 120000 iterations. During the re-training, no data augmentation is involved, and the optimization is achieved after 47600 iterations. We select 423 images with small objects and 456 images with medium objects for testing from the PASCAL VOC and the MS COCO. The present model is evaluated on a computer with ubuntu16.04, Intel Core i5-7400 CPU, and Nvidia GeForce GTX1060 GPU with graphic memory of 6GB. In the rest of this section, we visualize the detection results of our model for small and medium objects and compare the results with SSD. Furthermore, comparisons of overall performance on timing and accuracy between the present model, SSD, and other methods are made.

B. ABLATION STUDY

We investigate the effectiveness of different components of our model by the ablation study. The experimental results are shown in Table 1 and Table 2. ‘‘Deconvolution’’ refers to introduce a deconvolutional layer to magnify the low-resolution feature map. The deconvolution helps to improve the mAP of the small and medium object detection of SSD for 14.4% and 6.6%, respectively, because the larger feature maps may lead to higher classification accuracy for small and medium object detection. ‘‘Region proposal’’ means that we select the exiting layers in SSD that play a role of the region proposal network and the proposed regions are magnified for recognition. By adding the ‘‘Deconvolution’’ and ‘‘Region proposal’’, the model performance increase from 15.3% mAP to 34.2% mAP for small objects as shown in Table 2. Moreover, our model increases the performance by 8.2% for the medium objects as shown in Table 1. ‘‘Feature pyramid’’ represents a new feature pyramid that is constructed by the shallow layers. Since the feature map generated by the newly created feature pyramid not only preserves the features of small objects but also improves the accuracy of the object classification, the mAP is improved by 20.8% and 10.4% for the small and medium objects, respectively.

C. RESULTS AND DISCUSSIONS

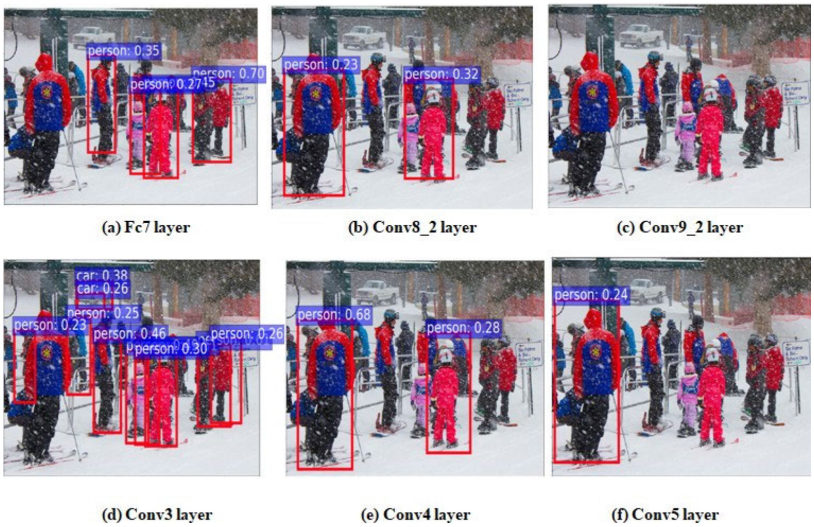
We illustrate some detection examples of specific layers in Fig. 7 and Fig. 8. For a covered object car in Fig. 7, the detection result corresponding to layer *conv*4_3 in SSD is plotted in the right panel, and the left panel shows the detection result corresponding to layer *Deconv*2 in the present model. It is shown that the bounding box given by SSD does not match the object size exactly, and the corresponding confidence is 0.02. However, the output of the present model not only matches the correct object size but also gives a confidence increasing nearly 6×. For dense objects in Fig. 8, i.e. peoples in the picture, the detection results of SSD are compared with the present model. Here, panels (*a*), (*b*), and (*c*) correspond to layers *Fc*7, *conv*8_2, and *conv*9_2 in SSD, and panels (*d*), (*e*), and (*f*) correspond to layers *conv*3, *conv*4, and *conv*5 in the present model. Notice that the dense medium object detection due to the layers *Fc*7, *conv*8_2, and *conv*9_2 is not satisfactory. There are numerous people that are not recognized by the detector. As can be



(a) SSD

(b) Present model

FIGURE 7. **(Color online) Detection results for a covered object generated by low-level feature maps. Here, the left panel shows the result corresponding to layer *conv*4_3 in SSD, and result corresponding to layer *Deconv*2 in the present model is provided in the right panel.**



(a) Fc7 layer

(b) Conv8_2 layer

(c) Conv9_2 layer

(d) Conv3 layer

(e) Conv4 layer

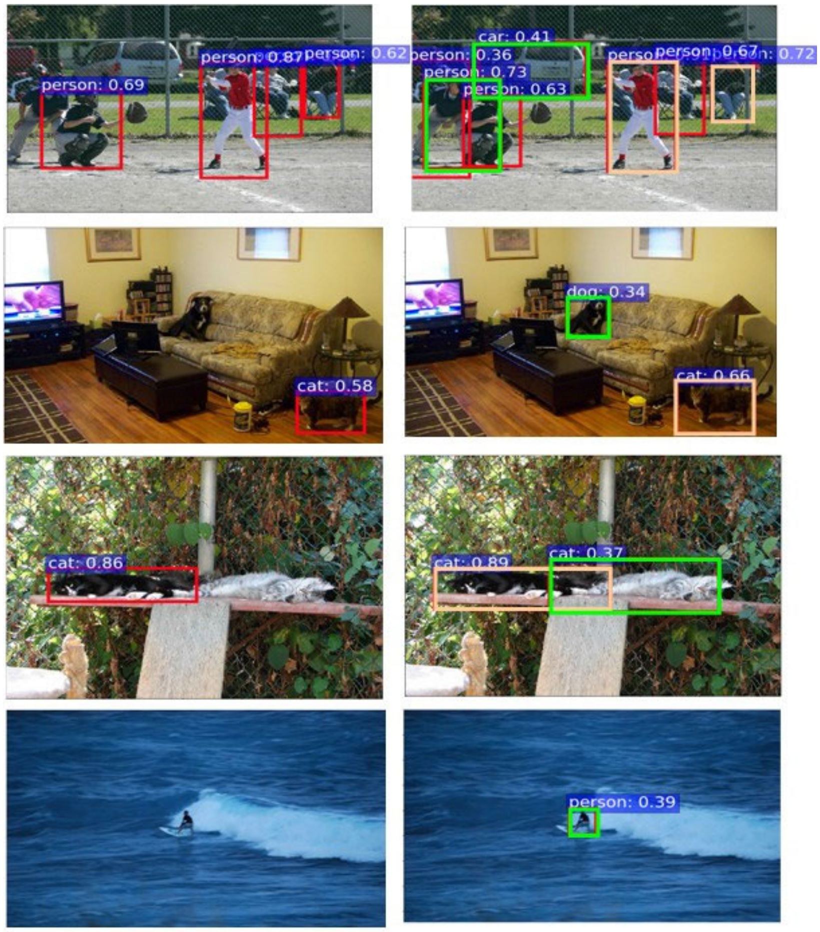
(f) Conv5 layer

FIGURE 8. **(Color online) Detection results for dense objects generated by high-level feature maps. Here, panels (a), (b), and (c) correspond to layers *Fc*7, *conv*8_2, and *conv*9_2 in SSD, and panels (d), (e), and (f) correspond to layers *conv*3, *conv*4, and *conv*5 in the present model.**

seen in panels (d), (e), and (f), the present model works remarkably well for recognizing those objects especially when peoples are shaded from others or far away from the camera.

Examples of detecting output of SSD and present model with scores higher than 0.3 are demonstrated in Fig. 9. Here, green bounding boxes represent the objects that are recognized by the present model but missed by SSD, yellow bounding boxes represent fault recognition by SSD, and purple bounding boxes represent the recognitions of the present model that have higher confidence than SSD. Fig.9 nicely proves the effectiveness of improvements we made that indeed result in better discrimination of the detector.

The average precision (AP) for a specific category and the mAP for all categories are effective metrics to evaluate object recognition models. We calculate the AP and mAP of the present model and other models, and the results are shown in Table 3 and Table 4. According to the code in literature [14], we reproduce a very recently developed version of SSD, i.e. FSSD. This model introduces an elaborately designed feature fusion module in which features from different layers in the feature pyramid are concentrated. A new feature pyramid is then established by pooling the concentrated feature maps to various sizes. Each of these feature maps may include vital location and semantic information. Using the same training strategy and testing data, we obtain the detection results of several other state-of-the-art SSD-based object detectors that are for object detection across different scales are also provided in Table 3 and Table 4. Here, outputs for medium objects are provided in Table 1 with the confidence



(a) SSD

(b) Present model

FIGURE 9. (Color online) Detection examples in our testing dataset with SSD and present model. Here, bounding boxes of different colors correspond to various detection outputs: green for missing recognitions by SSD, yellow for fault recognitions by SSD, and purple for recognitions that have higher confidence than SSD.

TABLE 3. Average precision of medium object detection for every category and mean average precision (mAP) for all categories. Here, the confidence threshold is taken as 0.3.

Model	bicycle	bus	car	cat	dog	motorbike	person	mAP
SSD	63.6%	63.6%	68.9%	43.2%	59.5%	57.8%	57.3%	59.1%
DSSD	70.8%	68.2%	73.3%	59.8%	74.2%	68.2%	58.6%	67.6%
FSSD	71.4%	59.4%	69.5%	60.3%	77.6%	77.9%	59.2%	67.9%
TDFSSD [21]	73.6%	74.2%	84.2%	75.5%	69.2%	72.3%	54.5%	71.9%
AugFPN [22]	74.3%	69.6%	95.2%	74.2%	78.6%	69.6%	57.2%	74.1%
Present model	79.5%	79.5%	98.3%	78.8%	62.7%	68.3%	59.2%	74.7%

TABLE 4. Average precision of small object detection for every category and mean average precision (mAP) for all categories. Here, the confidence threshold is taken as 0.1.

Model	bicycle	bus	car	cat	dog	motorbike	person	mAP
SSD	45.5%	3.0%	4.4%	26.0%	2.6%	13.6%	11.9%	15.3%
DSSD	47.6%	10.9%	15.2%	45.6%	36.1%%	53.6%	17.8%	32.4%
FSSD	43.2%	11.6%	14.9%	46.4%	38.0%	54.5%	19.6%	32.6%
TDFSSD	53.3%	22.6%	19.3%	49.2%	49.3%	61.9%	19.2%	39.3%
AugFPN	51.6%	20.3%	16.7%	51.8%	58.9%	62.2%	18.5%	40.0%
Present model	66.7%	21.4%	17.3%	52.2%	54.6%	67.5%	17.4%	42.4%

TABLE 5. Comparison of true positive rate (TPR) for class person between different models.

Model	SSD	DSSD	FSSD	TDFSSD	AugFPN	Present model
TPR	74.5%	77.6%	76.0%	81.2%	84.8%	84.5%

TABLE 6. Comparison of detection speed (measured in FPS) between different models.

Model	SSD	DSSD	FSSD	TDFSSD	AugFPN	Present model
FPS	55	21.5	48	19.4	12.1	24

higher than 0.3, and Table 4 displays small object detection with the confidence higher than 0.1. From Table 3 we note the obvious increase in the values of mAP. Particularly, the mAP for medium object detection of the present model increases by 15.6% compared to SSD. In addition, the results of our model are also higher than other methods. A similar tendency can also be found in the small object detections as shown in Table 4, where the mAP increases by 27.1% compared to SSD. Even though compared with the recent state-of-the-art model TDFSSD, our model's mAP exceeds the TDFSSD by 3.1%.

To examine the sensitivity for a specific category such as pedestrians, from MS COCO we pick up 1100 pictures including persons in various sizes and positions. Using these pictures, we calculate the true positive rate (TPR) which is defined as

$$TPR = \frac{TP}{TP + FN} \tag{8}$$

where TP is the number of true positive recognitions and FN is the number of false negative recognitions. The result is listed in Table 5. Notice that the TPR of our model has increased by 10.0% compared to SSD. Our model takes the first place in the state-of-the-art methods except AugFPN.

The detection speed is another critical criterion for an object detector that may have potential application in a real-time system, and it is often measured in FPS. Table 6 shows the comparison of detection speed between SSD, our model, and the other models on our testing environment. It is shown that FSSD can run at 48 FPS little slower than SSD that can run at 55 FPS. However, our model has a speed drop relative to previous models. The FPS of the present model is 24. For a single image, our model consumes more time twice than the previous models. As a matter of fact, speed vs. accuracy is the main trade-off of object detectors. The newly added layers that help to improve the performance on accuracy also deepen the network. With these layers, the model needs more calculation during forward and backward propagations, resulting in poor performance on training and inference speed. Since a typical video frame stream is usually 25 FPS, the present model could still satisfy the requirement of real-time detection.

IV. CONCLUSION

In summary, we have added a series of improvements on the existing SSD architecture. These improvements manage to increase the accuracy on small and medium object detections significantly over previous attempts. Our improvements include using an extra shallow layer in the backbone network, using a deconvolutional region magnification procedure to magnify low-level feature maps, and constructing a new feature pyramid on top of the existing SSD structure. With these modifications, we can achieve a dramatically improved performance much better than SSD even training on a dataset with a small number of samples. However, the newly added layers consume a lot of time resulting in the speed dropped by half, whereas it is still fast enough for real-time applications. In the future, it is worth to enhance our model with much stronger backbone networks such as ResNet [23] and DenseNet [24], and design a lightweight version of the model that is more appropriate for embedded systems.

REFERENCES

[1] G. E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[3] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," 2013, *arXiv:1312.6229*. [Online]. Available: http://arxiv.org/abs/1312.6229

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, Columbus, OH, USA, Jun. 2014, pp. 580–587.

[5] R. Girshick, "Fast R-CNN," in *Proc. ICCV*, Santiago, Chile, Dec. 2015, pp. 1440–1448.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[7] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," 2016, *arXiv:1605.06409*. [Online]. Available: http://arxiv.org/abs/1605.06409

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. ECCV*, Amsterdam, The Netherlands, Oct. 2016, pp. 21–37.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[10] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 6517–6525.

[11] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," 2017, *arXiv:1701.06659*. [Online]. Available: http://arxiv.org/abs/1701.06659

[12] J. Jeong, H. Park, and N. Kwak, "Enhancement of SSD by concatenating feature maps for object detection," 2017, *arXiv:1705.09587*. [Online]. Available: http://arxiv.org/abs/1705.09587

[13] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, "DSOD: Learning deeply supervised object detectors from scratch," 2018, *arXiv:1809.09294*. [Online]. Available: http://arxiv.org/abs/1809.09294

[14] Z. Li and F. Zhou, "FSSD: Feature fusion single shot multibox detector," 2018, *arXiv:1712.00960*. [Online]. Available: http://arxiv.org/abs/1712.00960

[15] K. S. Chahal and K. Dey, "A survey of modern object detection literature using deep learning," 2018, *arXiv:1808.07256*. [Online]. Available: http://arxiv.org/abs/1808.07256

[16] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," 2018, *arXiv:1809.02165*. [Online]. Available: http://arxiv.org/abs/1809.02165

[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 936–944.

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM MM*, New York, NY, USA, Nov. 2014, pp. 675–678.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Miscrosoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.

[20] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, Sardinia, Italy, 2010, pp. 249–256.

[21] H. Pan, J. Jiang, and G. Chen, "TDFSSD: Top-down feature fusion single shot multibox detector," *Signal Process., Image Commun.*, vol. 89, Nov. 2020, Art. no. 115987.

[22] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, "AugFPN: Improving multi-scale feature learning for object detection," in *Proc. CVPR*, Jun. 2020, pp. 12595–12604.

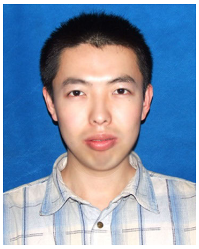
[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, Las Vegas, NV, USA, Jul. 2017, pp. 4700–4708.

YIZHONG WANG received the B.E. degree in optical instrument and the M.S. and Ph.D. degrees in measuring and testing technology and instrument from Tianjin University, Tianjin, China, in 1984, 1993, and 1996, respectively.

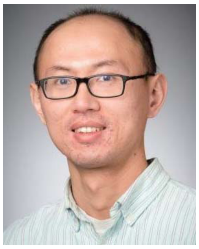
He was a Research Associate with the Department of Mechanical Engineering, The University of Hong Kong, Hong Kong, from 1997 to 2000 and from 2003 to 2004, and a Postdoctoral Fellow with the Department of Mechanical Engineering, Southern Methodist University, Dallas, TX, USA, from 2000 to 2012. Since 2004, he has been a Professor with the College of Electronic Information and Automation, Tianjin University of Science and Technology, Tianjin. His research interests include deep learning, computer vision, and the Internet of Things.

PENGHUI NIU received the B.E. degree from Shanghai Dianji University, Shanghai, China, in 2017. He is currently pursuing the M.S. degree with the Tianjin University of Science and Technology, Tianjin, China. His main research interests include deep learning and small object detection.



XIAOYONG GUO received the Ph.D. degree from Nanjing University, in 2012. He is currently an Associate Professor with the Department of Robotics Engineering, Tianjin University of Science and Technology. His main research interests include deep learning and data mining.

GUOWEI YANG received the B.E. and Ph.D. degrees from Tianjin University, Tianjin, China, in 2010 and 2015, respectively. His main research interests include deep learning and vision measurement.



JUN CHEN (Senior Member, IEEE) received the B.E. degree in communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2001, and the M.S. and Ph.D. degrees in electrical and computer engineering from Cornell University, Ithaca, NY, USA, in 2004 and 2006, respectively.

He was a Postdoctoral Research Associate with the Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, Urbana, IL, USA, from September 2005 to July 2006, and a Postdoctoral Fellow with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA, from July 2006 to August 2007. Since September 2007, he has been with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada, where he is currently a Professor. His research interests include information theory, machine learning, wireless communications, and signal processing.

Dr. Chen was a recipient of the Josef Raviv Memorial Postdoctoral Fellowship in 2006, the Early Researcher Award from the Province of Ontario in 2010, the IBM Faculty Award in 2010, the JSPS Invitational Fellowship in 2020, and the ICC Best Paper Award in 2020. He held the title of the Barber-Gennum Chair in Information Technology from 2008 to 2013 and the Joseph Ip Distinguished Engineering Fellow from 2016 to 2018. He served as an Associate Editor of the IEEE Transactions on Information Theory from 2014 to 2016. He is currently an Editor of the IEEE Transactions on Green Communications and Networking.