A Local Top-Down Module for Object Detection with Multi-scale Features

Shihua Huang, Lu Wang(✉), Peiyu Yang, and Qingxu Deng

School of Computer Science and Engineering,
Northeastern University, Shenyang, Liaoning, China
wanglu@cse.neu.edu.cn

Abstract. Object detection methods based on deep models and multi-scale features have achieved the state-of-the-art performance. However, since each feature layer operates independently, several issues such as box-in-box detections and less effective performance on small objects need to be addressed. In this paper, we tackle these issues by integrating contextual and semantic information from higher layer features into the prediction layer. Existing methods adopting similar ideas mostly apply full top-down modules, which may increase computational loads significantly. Instead, we present an efficient while general local top-down module, in which each prediction layer is integrated only with the upsampled features from its two succeeding layers. Experimental results show that the proposed algorithm performs favorably against the state-of-the-art methods on the VOC, COCO and HollywoodHeads datasets, while introducing little computational overhead. Compared with methods using full top-down modules, the proposed algorithm achieves comparable or higher accuracy while operates at a higher frame rate. The code is available at https://github.com/Hshihua/Local-Top-Down-Detection-Network.

Keywords: Object detection · SSD · Deconvolution
Local top-down module

1 Introduction

Object detection plays a pivotal role in image understanding that can be applied to numerous applications, e.g., image indexing and retrieval, image and video understanding, among others. Recent progress in object detection has been mainly based on deep Convolutional Neural Networks (CNNs). Existing CNN based object detection approaches can be categorized into two groups, based on image regions [1,14] or bounding boxes [10,11].

Among the state-of-the-art detection methods, the single shot multibox detector (SSD) approach [10] is fast and robust to scale variations because it makes use of multiple convolution layers for object detection. Although the SSD method performs well in terms of speed and accuracy, there are several issues that need to addressed. First, each layer in the feature pyramid of the SSD method is used independently as an input to the classifier network. Thus, the same object can be detected at multiple scales, which is known as box-in-box detections. Second, the first prediction layer has a smaller receptive field than others and thus has less context and weak semantic information. This is the main reason that the SSD method is less effective in detecting small objects.

In contrast to the existing bottom-up CNN models, top-down modules that forward information from top layers to lower layers have been shown to be critical for achieving the state-of-the-art performance. The top-down modules introduce abstract semantic information and context into lower level layers for better box classification [3,9,15] by continually upsampling features from the uppermost layer of the bottom-up pyramid to the lowest prediction layer. This operation does not stop until the finest resolution level of the feature pyramid has been reached. As such, approaches with such top-down modules may entail heavy computational loads.

In this paper, we show that it is possible to solve the problems of detectors using multi-scale feature maps by adding a simple and efficient Local Top-Down Module (LTD) that integrates the information from two succeeding convolutional layers instead of all the upper convolutional layers while introducing much less computational cost. For concreteness, we use the SSD detector as the baseline method with the proposed LTD module.

The contributions of this paper are summarized as follows:

– To solve the typical problems of detecting objects using multi-scale features, we propose a local top-down module to integrate high level semantic information and context into the lower prediction layers, which differs from the widely used full top-down module.
– Our method obtains significant improvement over the SSD method on different benchmark datasets. Compared with detectors using full top-down modules, our approach gets comparable or even higher accuracy.
– Our method runs at 37 FPS on a single 1080 GPU for images of 300×300 pixels, which is significantly faster than most existing detectors using full top-down modules.

2 Related Work

ConvNet Based Object Detectors. With significant progress of deep learning on large scale object recognition [4,5], numerous detection methods based on ConvNet have been proposed. Two-stage or region-based methods such as Faster R-CNN [14] and R-FCN [1] achieve high accuracy but with high computational loads. On the other hand, one-stage or region-free detectors such as YOLO [11, 12] and SSD [10] perform efficiently and accurately. The SSD [10] and YOLO9000 [12] methods do not entail the computationally expensive processes to generate region proposals while performing favorably against the state-of-the-art two-stage detectors. The recently proposed one-stage detector RefineDet [18] retains the merits of both one-stage and two-stage approaches by introducing an anchor refinement module that is able to filter out negative anchors as well as adjust the locations and sizes of anchors.

SSD Based Object Detectors. Numerous methods based on the SSD method have been developed. To generate high quality bounding boxes, Ren et al. propose the RRC scheme [13] by introducing a recurrent rolling convolution architecture over multi-scale feature maps to construct object classifiers and bounding box regressors. This method achieves the state-of-the-art performance on the KITTI dataset based on the IoU threshold of 0.7. Based on the observation that detecting small objects without enlarging the image requires more context information, the DSSD method [3] augments the ResNet [4] based SSD approach with deconvolution layers to introduce additional large-scale context. Recently, the R-SSD method [6] presents a feature concatenation structure that fully exploits the relationship among layers in the feature pyramid through pooling and deconvolution. By using the deconvolution module, both DSSD and R-SSD methods obtain about 1% performance gain over the SSD approach on the PASCAL VOC dataset.

Top-Down Module for Object Detection. Recent detection methods adopt top-down modules to add context and decode abstract but semantic information into low level feature maps for better box classification based on Faster R-CNN [14]. Lin et al. [9] propose a top-down model with lateral connections to construct semantic feature pyramid and achieve significant performance gain with a single model. In [8], Kong et al. use reverse connection with objectness of a prior network to combine fine-grained details with highly abstract information, which obtains performance gain in accuracy and speed. The TDM [15] method uses a top-down network that handles the selection and integration of contextual information and low level features. On the other hand, some recent methods are developed based on the SSD detector. The DSSD scheme [3] uses a deconvolution module with feature integration based on element-wise products. In the BlitzeNet [2] method, the ResSkip block is used to integrate feature maps from bottom-up and top-down streams with skip and residual connection. We note the above-mentioned methods all adopt full top-down module to construct top-down or backward feature pyramids.

3 Proposed Algorithm

Our goal is to improve the accuracy of object detectors using multi-scale features via a local top-down module. We implement our work based on the SSD method. Figure 1 illustrates the overall architecture of the proposed approach.

In the following, we first justify the necessity of using a local top-down module, and then describe the proposed local top-down module in details.

3.1 Main Ideas

The SSD method [10] is constructed based on a truncated base network that ends with several convolutional layers by adding a serial of progressively smaller

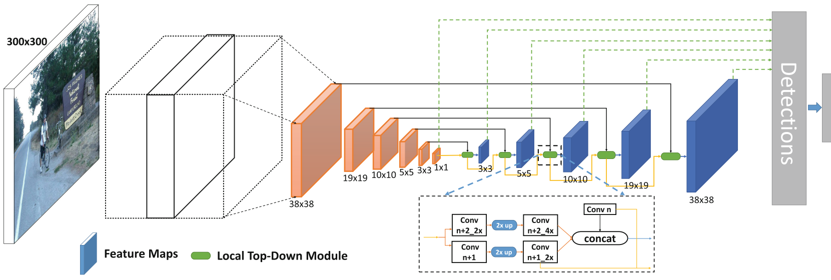

Fig. 1. Architecture of the proposed network.

convolutional layers. Each added layer together with some of the earlier base network layers are used to predict scores and corrections for the predefined default bounding boxes. However, the SSD method (or other detectors that utilize bottom-up multi-scale features) does not exploit the information between the lower layers and upper layers, e.g., the pathway that introduces more semantic information and context from high level features into low level ones. In particular, we show that not all the information propagated backward from the higher layers is useful for lower layers.

Considering the case that the input to a detector is an image of 300×300 pixels and an object to be detected is of 30×30 pixels, it is likely that the object can be detected on the 38×38 feature maps as there are much stronger activations in this layer than the others that can help locate this object effectively. However, in general the semantic information of such an object begins to vanish after the 10×10 feature maps ($300 : 30 = 10 : 1$), and the pixels within this object may then become the context of the other larger objects. Therefore, feature maps with resolution smaller than 10×10 are of little help in detecting this object. In other words, the most useful semantic information only exists in the next two higher layers (10×10 and 19×19 in this case) of the prediction layer (38×38 in this case). Furthermore, as the next two higher layers already have much larger receptive fields, they can provide sufficient context to the prediction layer.

Regarding the box-in-box detection issue, we observe that the scale of the outside box is usually at most 2 to 3 times the scale of the inside box, meaning that the visual information within three consecutive layers (4 times scale difference) is sufficient to solve this problem effectively.

As such, the effective top-down semantic and context information should be within local ranges (e.g., involving three feature layers), rather than the whole feature pyramid. Thus, we propose to use a local top-down module to propagate the upper layer information to the lower layers (e.g., within two to three layers).

3.2 Local Top-Down Module

Figure 2(a) shows our building block that constructs the local top-down feature maps for the n-th prediction layer, which are the concatenation of the bottom-up Conv n, upsampled Conv $n+1$ and Conv $n+2$ features. Specifically, the

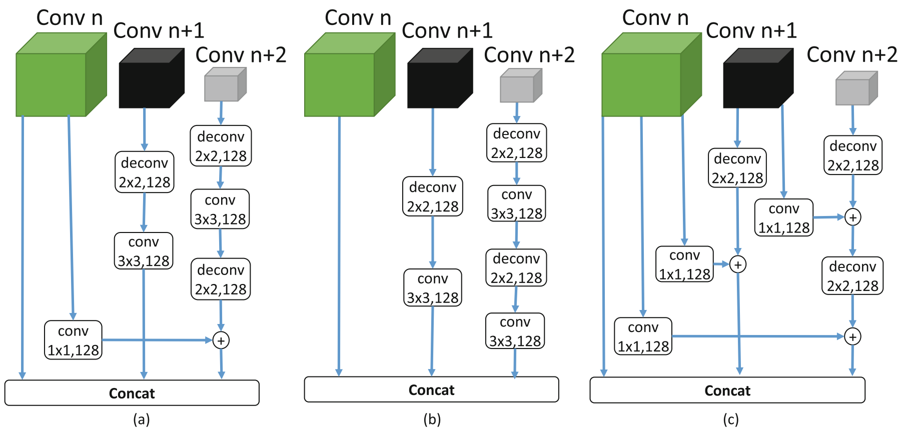

Fig. 2. Variants of the local top-down module. The difference lies in the block after deconvolution. (a) The convolutional block is integrated with the element-wise sum block (used in the proposed network); (b) all are convolutional blocks; (c) all are element-wise sum blocks.

Conv n+1 feature maps are first upsampled to the same spatial resolution as their pervious layer (Conv n) maps while being reduced to d channels via a learned deconvolution layer (d = 128 in our implementation). In case of information divergence and the aliasing effect of upsampling, the upsampled features are convolved with 3×3 kernels before being concatenated to the Conv n maps.

The Conv $n+2$ maps need to be upsampled twice before the concatenation. For the first deconvolution operation, they are processed in the same way as Conv n+1 maps (i.e., deconv + conv). The resulting features are used to construct the top-down maps of the Conv $n+1$ layer and then upsampled for the second time. Because the semantic information contained in the Conv $n + 2$ layer is diluted significantly after two continuous upsampling operations, the results are summed element-wisely with the Conv n maps (which undergo a 1×1 convolutional layer to reduce the channel dimension) to enhance the upampled features. The resulting features are finally concatenated to the bottom-up Conv n maps to form the local top-down feature maps for the n-th prediction layer.

In the proposed LTD module, for each prediction layer, the number of feature channels is increased for just two times that of the upsampled features, i.e., $2d$ (256 in our implementation). This is much smaller than that in the top-down module used by the R-SSD [6] method, which increases the feature dimension for up to 2304 channels. In addition, in our LTD module, the dimension of upper layer feature maps is reduced before they are merged with the prediction layer, whereas for the full top-down module, such as the DSSD scheme, feature channels need to be increased for certain layers before integration so as to make the element-wise product operation feasible through the whole feature pyramid. Hence, our LTD module is more efficient than these full top-down modules.

We also introduce two variants of the proposed LTD module to justify the design. For the first model, each deconvolution operation is followed by a convolution operation (Fig. 2(b)). For the second model, each deconvolution operation is followed by an element-wise addition with the corresponding bottom-up maps (Fig. 2(c)). Ablation studies with different local top-down modules are shown in Table 3 and discussed in Sect. 4.1.

3.3 Loss Function and Training

To illustrate the merits of the proposed local top-down module, we use the same settings as the SSD method including the data augmentation scheme, loss function, default boxes, and the means of matching default boxes to ground truth bounding boxes. We use activations of each layer in the local top-down pyramid to predict the class probability distribution and regress the coordinates of the default boxes.

4 Experimental Results

We evaluate the proposed algorithm against the state-of-the-art object detection methods on the PASCAL VOC 2007, PASCAL VOC 2012, MS COCO and HollywoodHeads datasets [17]. We also carry out ablation studies on the VOC 2007 dataset to better demonstrate the effectiveness of each component. All the hyperparameters are set according to the SSD method unless otherwise specified.

4.1 Results on the VOC 2007 Dataset

For the VOC 2007 dataset, all models are trained on the VOC 2007 trainval + 2012 trainval and evaluated on the VOC 2007 test set. The training process takes 120k iterations with initial learning rate being 10^{-3}, which is then decreased to 10^{-4} at 80k steps and to 10^{-5} at 100k steps. The detection performance of all evaluated methods is presented in Table 1. Overall, the proposed algorithm gains more than 2% improvement over the SSD method, and outperforms the others when the input image size is either 300×300 or 512×512 pixels.

We also carry out ablation studies to analyze the proposed method using the VOC 2007 datasets. The results are shown in Tables 2, 3 and 4 and discussed in the following. Unless specified explicitly, all the results are based on input images of 300×300 pixels.

Batch Normalization. Although batch normalization is a useful method in training deep models, it is not used in the SSD method. In this work, we add the batch normalization layer to convolutional layers from the last convolutional layer of the backbone until the last convolutional layer of the detection network. The experimental results show that this can increase the mAP for 0.8% on the VOC 2007 test set.

Backbone Network. We replace the VGGNet in SSD and our method with the MobileNet and ResNet-101 models as the backbone convolutional network to

Table 1. Detection results on the PASCAL VOC dataset. To exclude the effect of the different testing environments on the inference speed, the R-FPS metric is used to measure the relative processing speed of each SSD-based detector compared with that of the SSD detector running in the same environment. The letters M and P denote the Titan X Maxwell and Pascal architectures respectively. Bold fonts indicate the best

Method	Backbone	mAP(%)		FPS	R-FPS	GPU
		VOC 2007	VOC 2012			
two-stage:						
Faster R-CNN [14]	VGGNet	73.2	70.4	7		TITAN X(M)
R-FCN [1]	ResNet101	80.5	77.6	9		TITAN X(M)
DeNet-101 [16]	ResNet-101	77.0	73.9	33		Tesla P100
CoupleNet [19]	ResNet101	81.7	**80.4**	9.8		TITAN X(P)
one-stage:						
YOLOv2-352 [12]	Darknet	73.7	-	81	-	TITAN X(M)
SSD300 [10]	VGGNet	77.2	75.8	46	-	TITAN X(M)
R-SSD300 [6]	VGGNet	78.5	76.4	35	57%	TITAN X(P)
Blitznet300 [2]	ResNet-50	78.5	75.4	24	52%	TITAN X(M)
DSSD321 [3]	ResNet-101	78.6	76.3	15.3	33%	TITAN X(M)
YOLOv2-544 [12]	Darknet	78.6	73.4	40	-	TITAN X(M)
SSD512 [10]	VGGNet	79.8	78.5	19	-	TITAN X(M)
Blitznet512 [2]	ResNet-50	80.7	79.0	19.5	**103%**	TITAN X(M)
R-SSD512 [6]	VGGNet	80.8	-	16.6	66%	TITAN X(P)
DSSD513 [3]	ResNet-101	81.5	80.0	6.3	33%	TITAN X(M)
Ours300	VGGNet	79.4	76.7	37	85%	GTX1080
Ours512	VGGNet	**81.8**	79.7	16.7	87%	GTX1080

analyze the generalibility of the proposed algorithm. Table 2 shows the evaluation results with different backbone networks. Note that our method obtains at least 2% performance gain over the SSD method for different backbone networks. For the MobileNet, our method achieves about 4% gain. These results demonstrate the proposed method can be integrated with different backbone networks.

Variants of the LTD Module. To better understand the design options of the proposed local top-down module, we implement three variants as shown in Fig. 2. Table 3 shows the detection results of these variants. Note that the module with all element-wise sum blocks performs the worst while the proposed model based on convolution and element-wise performs best. These results show that the adopted model with convolution and summation operation performs better than the model with pure convolution or summation operations.

Number of Upsampled Convolutional Layers. Table 4 shows the detection results when different number of succeeding upsampled convolutional layers are integrated with the prediction layer. When incorporating three upsampled

Table 2. Detection performance with different pre-trained backbones.

Backbone	Method	mAP(%)
MobileNet [5]	SSD	68.3
	ours	**72.3**
VGGNet [7]	SSD	77.2
	ours	**79.4**
ResNet-101 [4]	SSD	73.3
	ours	**75.5**

Table 3. Comparison of detection performance with different local top-down modules.

Input size	LTD Module	mAP(%)
300×300	all-elementwise-sum	78.9
	all-conv	79.3
	conv-and-sum (Proposed)	**79.4**
512×512	all-elementwise-sum	81.4
	all-conv	81.5
	conv-and-sum (Proposed)	**81.8**

layers, the Conv $n+3$ feature maps additionally undergo a 3×3 convolution, a deconvolution and an element-wise sum operation after those processing steps for the Conv $n+2$ features. The results in Table 4 show that incorporating the upsampled feature maps from the two succeeding convolutional layers performs best, which justifies our design and proves that layers that are far away from the prediction layer does not help in achieving better detection performance.

To evaluate the inference time, we test the SSD method and ours on the VOC 2007 dataset with a single 1080 GPU. For input images of 300×300 pixels, the proposed method runs at 37 FPS while the SSD scheme runs at 43.6 FPS. For input images of 512×512 pixels, the proposed method runs at 16.7 FPS while the SSD scheme runs at 19.2 FPS. By introducing the local top-down module, the speed difference is no more than 15%. More run-time evaluation results with other methods are shown in Table 1, which demonstrates that our method introduces much less overhead than the other SSD based detectors, except for the BlitzNet512 detector. Note that the BlitzNet512 uses the ResNet50 as the backbone, which is 2 times faster than the VGG backbone used by us, and the mAP of BlitzNet is 1.1% lower than our method.

Table 4. Results for integration with different number of succeeding upsampled convolutional layers.

# upsampled layers	0	1	2	3
mAP(%)	77.2	79.1	**79.4**	78.7

4.2 Results on the VOC 2012 Dataset

We also test our method on the PASCAL VOC 2012 dataset. For VOC 2012 test set, we train our network on data consisting of PASCAL VOC 2007 trainval, 2007 test and 2012 trainval. The training process takes 240k iterations with initial learning rate being 10^{-3}, which is decreased to 10^{-4} at 160k steps and to 10^{-5} at 200k steps.

The detection results are shown in Table 1. We see that our method gains 0.9% and 1.2% improvement over the SSD method and compares favorably against the other state-of-the-art object detectors. Note that the performance gap between our method and the DSSD method becomes smaller than that on the PASCAL VOC 2007 test set. Specifically, when the input is about 300×300 pixels, the mAP gap between our method and the DSSD scheme is 0.4%, while the gap is 0.8% on the VOC 2007 dataset; When the input is around 512×512 pixels, the DSSD method outperforms our method by 0.3%. We think the main reason behind this is that DSSD is based on the deeper ResNet101 [4] backbone and the complicated classifiers, which can benefit more from the larger amount of training data than the VGGNet [7] used in our method. This assert can be justified by the fact that the SSD detector with the ResNet101 backbone performs worse on PASCAL VOC 2007 test set than the SSD with the VGGNet backbone, but better on PASCAL VOC 2012 test set and MS COCO test-dev2015 data.

4.3 Results on the MS COCO Dataset

We conduct experiments on the MS COCO 2015 dataset where the models are trained on the trainval35k set and evaluated on the test-dev. The training process takes 400k iterations with the initial learning rate being 10^{-3}, which is decreased at 280k and 360k steps to 10^{-4} and 10^{-5} respectively. We report the standard COCO metrics including AP (averaged over IoU thresholds), AP_{50} and AP_{75} (measured under IoU threshold of 0.5 and 0.75 respectively). In addition, we present the COCO AP on objects of small, medium, and large size (namely, AP_S, AP_M, and AP_L).

Table 5 shows the detection performance of different detectors on the MS COCO dataset. Note that both SSD321 and DSSD methods achieves better mAP than our algorithm. However, most the performance gain is from the deeper ResNet-101 backbone network and strong classifiers. When comparing the SSD321 with the DSSD321, although the deconvolution module of DSSD achieves 1.2% improvement for small object detection, it does not improve the overall accuracy of SSD, i.e., the performance of DSSD is inferior to SSD in detecting medium- and large- size objects. On the other hand, our method consistently outperforms the SSD scheme significantly under all the evaluation conditions.

4.4 Results on the HollywoodHeads Dataset

As heads usually correspond to small objects in images and head detection is useful for various applications, we evaluate our method on the HollywoodHeads

Table 5. Detection performance on the MS COCO test-dev2015.

Method	Training data	Backbone	Input Size	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
two-stage:									
Faster-RCNN [14]	trainval	VGGNet	$\sim 1000 \times 600$	21.9	42.7	-	-	-	-
R-FCN [1]	trainval	ResNet101	$\sim 1000 \times 600$	29.9	51.9	-	10.8	32.8	45.0
DeNet-101 [16]	trainval35k	ResNet101	512×512	31.9	50.5	34.2	9.7	34.9	50.6
CoupleNet [19]	trainval	ResNet101	$\sim 1000 \times 600$	33.1	53.5	35.4	11.6	36.3	50.1
one-stage:									
YOLOv2 [12]	trainval35k	DarkNet	544×544	21.6	44.0	19.2	5.0	22.4	35.5
SSD300 [10]	trainval35k	VGGNet	300×300	25.1	43.1	25.8	6.6	25.9	41.4
SSD321 [3]	trainval35k	ResNet101	321×321	28.0	45.4	29.3	6.2	28.3	49.3
DSSD321 [3]	trainval35k	ResNet101	321×321	28.0	46.1	29.2	7.4	28.1	47.6
SSD512 [10]	trainval35k	VGGNet	512×512	28.8	48.5	30.3	10.9	31.8	43.5
SSD513 [3]	trainval35k	ResNet101	513×513	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	trainval35k	ResNet101	513×513	**33.2**	53.3	35.2	13.0	35.4	51.1
Ours300	trainval35k	VGGNet	300×300	27.5	47.7	28.1	**9.2**	28.7	43.0
Ours512	trainval35k	VGGNet	512×512	31.3	53.1	32.5	**14.3**	33.5	45.4

Table 6. Experimental results on the HollywoodHead test set.

Method	Training data	mAP(%)
Baseline [17]	training set	72.7
SSD300 [10]	validation set	70.0
Ours 300	validation set	**75.6**

dataset [17]. This dataset contains 369, 846 human heads annotated in 224, 740 frames from 21 Hollywood movies. The movies vary in genres and contain different time epochs. The training set contains 216, 719 frames, the validation set consists of 6, 719 frames and the test set is composed of 1,302 frames.

In this experiment, we use the validation set as the training set to train SSD and our method, as the original training set is large that either determining the proper train parameters or the training process itself is computationally prohibitive. We set the learning rate to be 0.01 for the first 16 k iterations, 10^{-4} for the next 4k iterations and 10^{-5} for the last 4k iterations. Table 6 shows that our method outperforms both the detectors in [17] and the SSD method, even when the detector in [17] is trained on a much larger training set.

4.5 Visual Illustration of the Detection Results

Figure 3 shows some examples of the detection results of SSD and our algorithm, when the threshold of the detection confidence score is 0.6 for both methods. The first two rows show the detection results on the VOC and MS COCO datasets

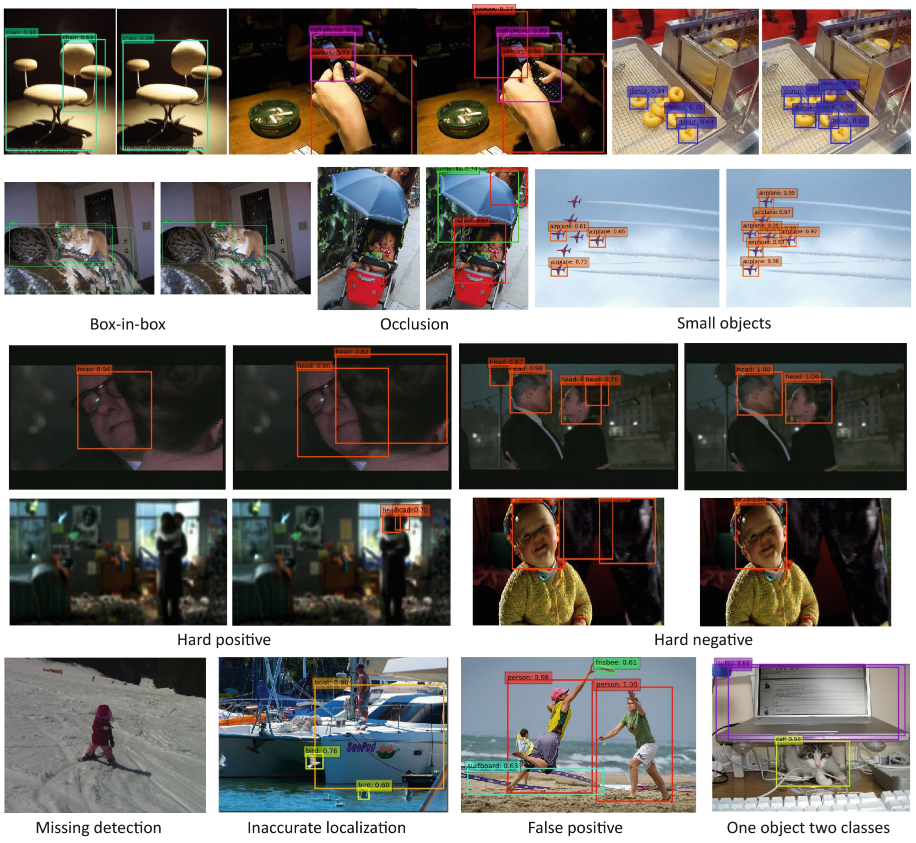

Fig. 3. Illustration of the detection results. The first and second rows show some improved results of our method (right) over SSD (left) on the VOC, COCO and HollywoodHeads datasets. The last row shows some failure cases of our method.

and next two rows display the results on the HollywoodHeads datasets. Overall, the proposed LTD method can deal with some difficult detection cases such as box-in-box, small objects and occlusion better than the SSD scheme. The last row of Fig. 3 illustrates some failure cases of our method. There are occasional missing detections, inaccurate box localizations and false positives caused by various reasons. The missing and inaccurate detections in the first two images are mainly due to the ambiguous appearance of the objects. In the third failure case, the false positive detection occurs due to the fact that the shape formed by the belt happens to look like a surfboard while the person on top of it and the sea form a reasonable context to support its existence. In the fourth image, the laptop is also detected as a TV due to the similarity between the two classes.

5 Conclusion

In this paper, we present a local top-down module to solve the problems of object detectors with multi-scale features. Although the proposed approach is simple, i.e., only concatenating the features upsampled from the two succeeding convolutional layers with the current prediction layer, it is effective and efficient in achieving performance gain on large datasets based on the SSD method. The proposed LTD module can be applied to other detection frameworks that utilize multi-scale features. Our future work is to incorporate the proposed LTD module into other state-of-the-art object detectors such as RON [8].

Acknowledgements. This work is supported by Natural Science Foundation of Liaoning Province, China, #20170540312.

References

1. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: NIPS, pp. 379–387 (2016)
2. Dvornik, N., Shmelkov, K., Marial, J., Schmid, C.: BlitzNet: a real-time deep network for scene understanding. In: ICCV, pp. 4174–4182 (2017)
3. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: DSSD: deconvolutional single shot detector. arXiv preprint arXiv: 1701.06659 (2017)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
5. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv: 1704.04861 (2017)
6. Jeong, J., Park, H., Kwak, N.: Enhancement of SSD by concatenating feature maps for object detection. In: BMVC (2017)
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
8. Kong, T., Sun, F., Yao, A., Liu, H., Lu, M., Chen, Y.: RON: reverse connection with objectness prior networks for object detection. In: CVPR, pp. 5244–5252 (2017)
9. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: ICCV, pp. 936–944 (2017)
10. Liu, W., et al.: SSD: Single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
11. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: CVPR, pp. 779–788 (2016)
12. Redmon, J., Farhadi, X.: YOLO9000: better, faster, stronger. In: CVPR, pp. 6517–6525 (2017)
13. Ren, J., et al.: Accurate single stage detector using recurrent rolling convolution. In: CVPR, pp. 752–760 (2017)
14. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal network. In: NIPS, pp. 91–99 (2015)
15. Shrivastava, A., Sukthankar, R., Malik, J., Gupta, A.: Beyond skip connections: top-down modulation for object detection. arXiv preprint arXiv: 1612.06851 (2016)

16. Tychsen-Smith, L., Petersson, L.: Denet: scalable real-time object detection with directed sparse sampling. In: ICCV, pp. 428–436 (2017)
17. Vu, T.H., Osokin, A., Laptev, I.: Contex-aware CNNs for person head detection. In: ICCV, pp. 2893–2901 (2015)
18. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. In: CVPR (2018)
19. Zhu, Y., Zhao, C., Wang, J., Zhao, X., Wu, Y., Lu, H.: Couplenet: coupling global structure with local parts for object detection. In: ICCV, pp. 4146–4154 (2017)