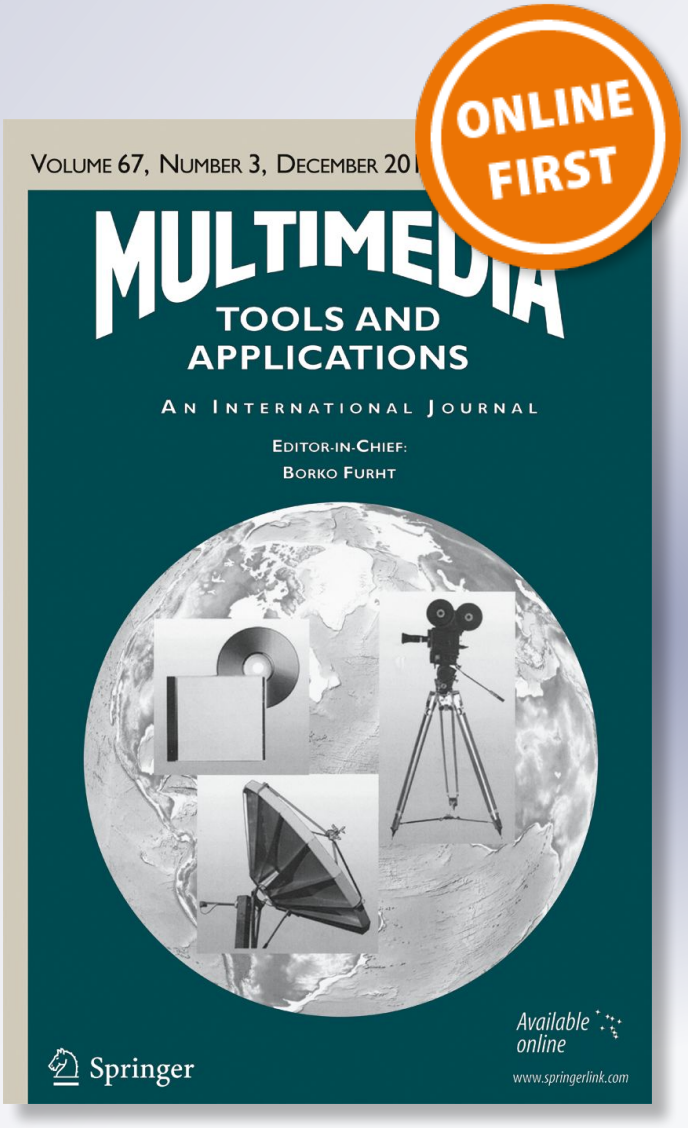
Small-objectness sensitive detection based on shifted single shot detector

Liangji Fang, Xu Zhao & Shiquan Zhang

Multimedia Tools and Applications
An International Journal

ISSN 1380-7501

Multimed Tools Appl
DOI 10.1007/s11042-018-6227-7



Springer

Multimed Tools Appl
https://doi.org/10.1007/s11042-018-6227-7

Small-objectness sensitive detection based on shifted single shot detector

Liangji Fang[1] · **Xu Zhao**[1] · **Shiquan Zhang**[1]

Received: 28 July 2017 / Revised: 27 December 2017 / Accepted: 29 May 2018

Abstract We present a small object sensitive method for object detection. Our method is built based on SSD (Single Shot MultiBox Detector (Liu et al. 2016)), a simple but effective deep neural network for image object detection. The discrete nature of anchor mechanism used in SSD, however, may cause misdetection for the small objects located at gaps between the anchor boxes. SSD performs better for small object detection after circular shifts of the input image. Therefore, auxiliary feature maps are generated by conducting circular shifts over lower extra feature maps in SSD for small-object detection, which is equivalent to shifting the objects in order to fit the locations of anchor boxes. We call our proposed system Shifted SSD. Moreover, pinpoint accuracy of localization is of vital importance to small objects detection. Hence, two novel methods called Smooth NMS and IoU-Prediction module are proposed to obtain more precise locations. Then for video sequences, we generate trajectory hypothesis to obtain predicted locations in a new frame for further improved performance. Experiments conducted on PASCAL VOC 2007, along with MS COCO, KITTI and our small object video datasets, validate that both mAP and recall are improved with different degrees and the speed is almost the same as SSD.

Keywords Object detection · Shifted SSD · Smooth NMS · IoU prediction

✉ Xu Zhao
zhaoxu@sjtu.edu.cn

Liangji Fang
fangliangji@sjtu.edu.cn

Shiquan Zhang
15221529981@163.com

[1] Department of Automation, Shanghai Jiao Tong University, Shanghai, China

Published online: 14 June 2018

1 Introduction

For traditional computer vision tasks, hand-crafted features such as HOG [39] and SIFT [28] may be less robust to the variances in the objects. The combination of these hand-crafted features causes the increasing of the dimension of data feature space which is inefficient. Feature selection method [23] only uses a subset of the original features and [22] learns a robust representation of data from ground truth information using the latent data structure. Recently, deep learning, which learns feature representation from data automatically, has been wildly used in domains of image classification [15, 36], semantic segmentation [27, 42], human pose recovery [18, 19, 43] and other computer vision fields [44]. In the context of deep neural networks for object detection, the current state-of-the-art methods could be roughly divided into region proposals based methods [10, 11, 13, 14, 34, 41] or direct regression based methods like YOLO [32, 33], SSD [25] and [21, 38, 46]. The region proposals based methods divide the task into two stages: the first is proposal generating process and the second is classification and localization process for proposals. The straightforward regression methods directly regress the class and location of some fixed predefined boxes. Although these two kinds of methods work well on large objects, they perform unsatisfyingly on small object detection. When it comes to small object detection, the region proposals based methods tend to increase the size of input images [3], add context information [45]. It may achieve higher accuracy, but the sacrifice of speed is intolerable especially based on the two stages paradigm. Although the direct regression based methods are extremely fast, they may struggle with relatively worse performance. Compared with other methods, SSD achieves high accuracy and fast speed at the same time. However, it is limited in detecting small objects and we do not know exactly what the problems are and how to solve them.

In this paper, we first investigate why SSD does not perform well on detecting small objects and three main problems are observed: (a) Because of anchor box mechanism, SSD only obtains the weak translation invariance for small object detection. If the objects move slightly in the image, the detection results will change accordingly. (b) When detecting small objects, SSD struggles with outputting more precise locations. The most precisely localized boxes generated by SSD may not have the highest confidences. Hence after the post-process like non-maximum suppression (NMS), these precisely located boxes will be suppressed. (c) SSD has a severe overfitting issue for detecting small objects.

To solve these problems, we propose Shifted SSD for small-object detection. Firstly, we find that the weak translation invariance issue is eased by circularly shifting the input image before feeded to SSD. To reduce the computation redundancy, we shift the lower level feature maps circularly for prediction instead of shifting the input images. Then we combine the detections produced by shifted feature maps and the original detections from SSD to get the final results. Secondly, to obtain more precise locations for small objects, we propose two novel methods named as Smooth NMS (SNMS) and IoU-Prediction respectively. The post-process called NMS in SSD suppresses a lot of accurately localized detections with slightly lower confidence scores. Therefore, SNMS utilizes these accurately localized detections to generate finer results. While only classification information is regarded as confidence score for NMS in SSD, IoU-Prediction provides localization accuracy information which is combined with classification information for NMS or SNMS. Lastly, we apply our Shifted SSD to object detection in videos as an extension. When SSD is used for small-object detection in videos, it will miss some objects within every few frames which are detected in the previous frames. We notice that some detections whose confidences are

lower than the threshold (weak detections) are also positive results. Therefore, another main problem of miss-detection in videos is the prior setting confidence threshold. So we use trajectory hypothesis to reduce the threshold selectively and increase the continuity of object detection in videos.

Our main contributions can be summarized as follows:

(i) We quantitively analyze the problems of SSD in detecting small object and propose the Shifted SSD, which shifts feature maps to ease the impact of the discreteness of anchor boxes method.
(ii) The proposed Shifted SSD achieves a better performance along with the proposed SNMS and IoU-Prediction methods for more accurate localization. To further improve its capability to utilize temporal information, we introduce trajectory hypothesis to increase the continuity of object detection in videos.
(iii) Experiments conducted on PASCAL VOC 2007, KITTI, MS COCO databases along with our small object video database, validate the effectiveness of our proposed method.

The rest of this paper is organized as followings. We discuss the related work in Section 2 and analyze problems of SSD in detecting small objects in Section 3. Then in Section 4, our Shifted SSD is proposed to solve the issues mentioned in Section 3. In Section 5 we show how to use both weak and strong detections in sequential detection. Implementation details and experimental results are provided in Section 6. We conclude in Section 7.

2 Related work

The region proposals based method R-CNN [11] uses Selective Search to generate proposals, afterwards the image is cropped based on them and CNN is used to classify the proposals. To speed up, SPP-Net [13] and Fast R-CNN [10] use RoI pooling approach to compute feature maps only once for all the proposals generated from a single image. Furthermore, Faster R-CNN [34] design a architecture to generate region proposals and share the computational features with the classification CNN. In order to obtain more precise localization results, LocNet [9] design a network to predict conditional probabilities of each row and column in the marginal region of an object.

YOLO and SSD only look once on the input images and directly regress the class and location information, so they are extremely fast. Unlike region based methods that generate proposals using image informations, they rely on a set of predefined fixed boxes and predict their class and location information. As shown in [12, 27], lower-level feature maps capture more fine details of the objects which are more accurate for localization, while top layers extract high-level features and contain more global context information [26] which are beneficial for classification. SSD achieves higher accuracy and almost the same speed compared with YOLO because both low-level and high-level feature maps are utilized for detecting objects with different sizes without fully connected layers.

To the best of our knowledge, there are few region proposals based methods proposed for small object detection. [3] follows the R-CNN paradigm, adding more context information and using fine grained layer to generate region proposals compared to RPN [34], however it is extremely slow. [45] uses the à trous trick [4] and concatenates the features from different layers to solve the collapsing bins problem in small objects detection, which only runs at 0.5 fps.

3 Problems analysis

We first investigate why SSD does not perform well on detecting small objects (some quantitative results are shown in Fig. 1) and then carefully design experiments to quantify these disadvantages. Because SSD uses lower layers to detect smaller objects, we define small objects as that with area smaller than the area of the biggest anchor box on the lowest prediction layer (e.g. conv4_3).

3.1 Weak translation invariance

Similar to Faster RCNN [34], SSD uses anchor box mechanism to obtain a set of fixed default boxes, in which the input image is divided into grids according to the size of each feature maps used for prediction. The center of each cell represents the center of certain anchor boxes predicted on the corresponding feature map. SSD utilizes different layers for prediction and put multiple anchor boxes with different sizes on these prediction layers. To detect small objects, the scales of anchor boxes on lower feature maps should be set relatively small, which however will lead to a problem that if the scale is too small, the anchor boxes on a certain feature map can not cover the entire input image after mapping (Fig. 2) and the objects located in the gap between two default boxes couldn't be detected. Although the localization problem could be refined by regression, the improvement may be limited. Even if the anchor boxes could cover the input image, the probability of missing detection is still high when the objects locate near the grid lines. What if we shift the objects by half size of the cell so that the objects near the grid line will move to the center of the cell? in order to shift the objects to avoid the problems mentioned above and combine the original detections, we circularly shift the input image before input to SSD and the performances are saliently improved. Some examples corresponding to this problem are shown in Fig. 1. As mentioned in [34], anchor box mechanism is translation invariant, however we find that anchor box mechanism only obtain weak translation invariance for small object detection.

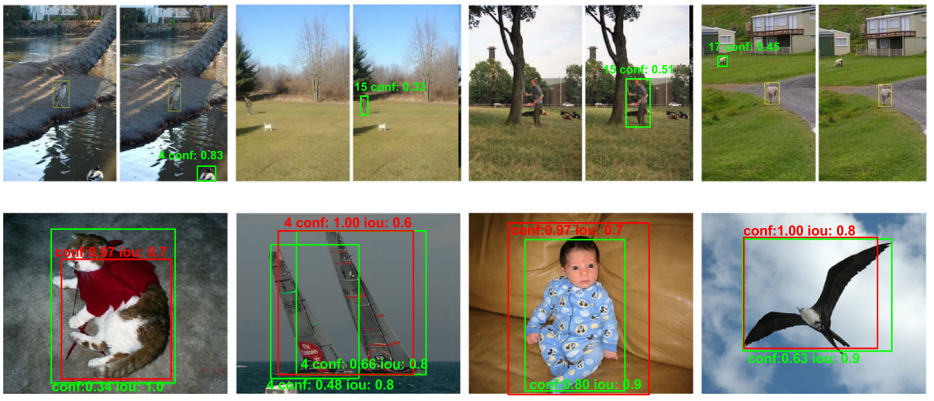


Fig. 1 Problems of SSD in detecting small objects on VOC 2007 dataset. **First row:** Examples of weak translation invariance. The left image of each pair represents the results of the original image and the right image represents the results of the circularly shifted image. The bold green bounding boxes are the targets detected in a image but missing in another image. Confidence score and iou matched with the ground truth are denoted as "conf" and "iou". **Second row:** Examples of Reverse out. Red bounding boxes are the outputs of SSD which have the highest confidences after NMS while green bounding boxes are the most accurate detections before NMS

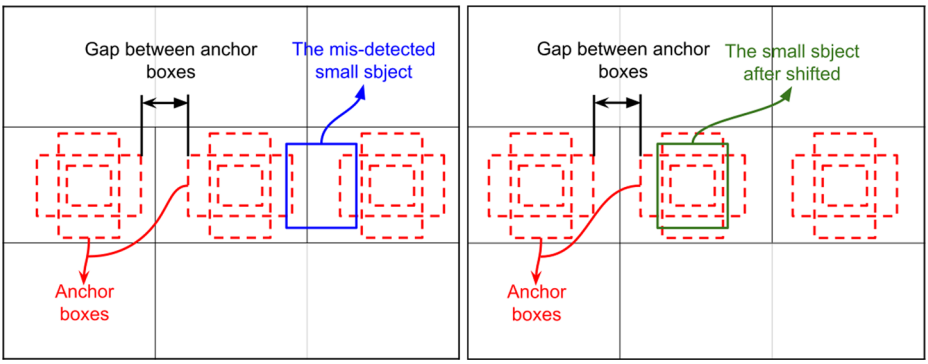


Fig. 2 Motivations of the proposed Shifted SSD. Left: the original image. Right: the image shifted by half size of the cell

To quantify the phenomenon mentioned above, we first circularly shift all the resized test images in Pascal VOC 2007 by 4 and 8 pixels[1] in four directions, notated as left, down, left up and right down[2] respectively. Then we input these images into the original SSD300* model. If the changing amplitude of confidence or IoU (intersection over union) of the detection is greater than a threshold (e.g 0.1), we consider that the detection result of this ground truth is changed significantly after shifting. Last, we calculate the ratio between the number of significantly changed detections and the number of ground truth objects.

Table 1 shows corresponding data on Pascal VOC 2007 that the translation invariance of small objects is weaker than big objects, which means that if the input images are shifted a bit then nearly 20% of the detection results will be changed significantly.

3.2 Reverse out

During training, SSD matches each ground truth object with the best overlapped anchor box and any anchor boxes whose IoU is larger than a threshold (e.g 0.5). Ideally, we would expect the best overlapped anchor box will have the best prediction during testing. However, based on our observation, the best overlapped anchor box usually has the highest confidence but not the most accurate location as shown in Fig. 1 bottom.[3] None maximum suppress (NMS) inhibits the detections with lower confidences which lead to relatively inaccurate outputs. To quantify this phenomenon, we output all the detections before NMS and find the detections with the most accurate locations. Then we compare them with the detections (which have the highest confidences) after NMS. If they are different for the same ground truth object, it means the output detections do not have the highest confidence and the most accurate locations. Last we count the number of the output detections whose confidences and locations are not the best at the same time.

Our experiment shows that almost 57% of the output detections do not have the most accurate locations among all the output detections and especially for small objects.

[1]For SSD300* model, the step between anchor boxes is 8 pixels on the lowest prediction layer and 16 on the next layer. So 4 and 8 pixels are the half length of the grid.

[2]For simplicity, left shifted direction is treated equivalently to right shifted direction.

[3]Small objects have little visualization differences in localization accuracy. For the clarity of visualization , we just illustrate examples of big objects on Reverse out phenomenon.

Table 1 Quantification of the **weak translation invariance** phenomenon

We calculate the Changed ratio after shifting the input images along four directions on VOC 2007 dataset. Shift step indicates the circularly shifted pixels while small_only means only small objects are considered

Shift step	Changed ratio			
	Shift Direction			
	left	down	left-up	right-down
4(small_only)	0.26	0.20	0.31	0.30
4(big_only)	0.20	0.15	0.24	0.25
8(small_only)	0.28	0.21	0.34	0.34
8(big_only)	0.20	0.15	0.25	0.26

3.3 Overfitting

SSD uses multiple layers for prediction and the lowest layer is responsible for the small objects. In Table 2 we summarize the approximate[4] performance of the lowest prediction layer in the original SSD on both Pascal VOC 2007 and KITTI datasets for small and big objects respectively. Note that for small-object detection, we only use the lowest prediction layer and all the prediction layers except for the lowest layer for big objects detection.

Table 2 shows that SSD performs well on big objects for both train and test sets. However, it performs poorly on small objects, especially on KITTI. The difference of mAP between train and test sets is over 70%. So we can say that SSD has severe overfitting problem on small-object detection.

4 Shifted SSD

Circular shift means all the rows or columns of feature maps are shifted to the next position and the final ones are moved to the first position. Direct circular shifting of the input image could reduce miss-detection. However, this will introduce redundant computation since all the convolution need to be computed once more. In this section, we present Shifted SSD model on lower level feature maps instead of the input image. We first discuss how to ease the problems we find in Section 2. Then, we show the model of Shifted SSD. Last we introduce the training procedure.

4.1 Which layers to shift

We first try to use denser boxes for small object detection to ease weak translation invariance by using smaller pooling stride. But the result is not satisfied (As shown in Table 4 column Stride). So we propose our Shifted SSD model which circular shift the feature maps to realise the same effect of circular shift input image as mentioned in Section 3.1. We establish two simple principles to determine which layers to circular shift.

– **Principle 1**. After shifting, new features should be generated. Where there is no new features, there is no new detections.

[4]Because we do not know the exact matching between ground truth target and the predictions of the lowest layer.

Table 2 Performance on both train and test phase on VOC 2007 and KITTI datasets

Dataset	Objects size	Phase	mAP	Overfit
VOC 2007	small	train	60.2%	23.7%
		test	36.5%	
	big	train	93.6%	10.6%
		test	83.0%	
KITTI	small	train	83.3%	65.1%
		test	18.2%	
	big	train	85.0%	9.9%
		test	75.1%	

Detections and ground truth targets are divided into small and big sets according to the area of their bounding boxes

– **Principle 2**. Operation should be on lower layers of the feature maps to detect small-object and reduce computation redundancy.

We build our model based on SSD which uses layers conv4_3, conv7, conv8_2, conv9_2, conv10_2, and conv11_2 to predict object locations and class confidences. Considering the principle 2, we use lower feature maps conv4_3 and conv7 to generate shifted layers. However, the convolution kernel size used to predict location and confidence is $3 \times 3 \times p$ and the stride is 1 on these two layers. Based on the principle 1, direct circular shift of the input of these layers will not generate new features and will get the same detections as just shifting locations. But if we circularly shift the input of a layer whose convolution stride is 2 or other even number by an odd number smaller than the stride, new features will be generated. Therefore, we can circularly shift the nearest convolution layers whose stride is 2 before conv4_3 and conv7. Then we replicate all the layers between them. The network architectures of the shifted SSD and SSD are shown in Fig. 4. What should be noted is that only shifted layers before conv4_3 are shown for simplicity.

4.2 How to get more precise location

According to Section 3.2, there are plenty of accurately localized detections with lower confidence scores generated by SSD, especially for small objects. Due to the NMS procedure, these accurate detections are suppressed. Thus we propose two methods called SNMS and IoU-Prediction to refine the final locations and confidence scores respectively.

Smooth NMS We first propose a modified NMS method called smooth NMS (SNMS) in which not only the detection with highest confidence but also detections with top k confidences which are suppressed by the detection with highest confidence are considered. The final output of smooth NMS is the average among them :

$$box_j = \frac{1}{k} \sum_{i=1}^{k} det_i, conf_i >= conf_j * thre \tag{1}$$

where box_j is the averaged location $[xmin, ymin, xmax, ymax]$ of the SNMS output, det_i are the detections $[xmin, ymin, xmax, ymax]$ suppressed by NMS res_j (including res_j), $conf_j$ is the confidence of NMS result res_j, and $thre$ is a threshold (e.g 0.7) selecting the detections suppressed by res_j.

IOU-Prediction To solve the problem that the detection with most accurate location but does not correspond to the highest confidence, we add another regression branch called

IoU-Prediction regression which is similar to the location and confidence regression. IoU-Prediction tries to predict the IoU between an output box of SSD and a ground truth box. The predicted IoU information will be used for choosing better detection boxes with both higher confidences and more accurate locations. A new score P^*_{det} is proposed for NMS or SNMS process and P^*_{det} is defined as

$$P^*_{det} = P_{cls} \cdot P_{IoU} \tag{2}$$

where P_{cls} is the confidence score for each box in original SSD and P_{IoU} is the predicted IoU for each box from the IoU-Prediction.

While softmax loss function is used for classification and smooth L1 loss function is used for localization, we use Euclidean loss function for IoU-Prediction which is defined as

$$L_{iou\_pred} = \frac{1}{2P} \sum_{i=1}^{P} \left\| IoU_i^{gt} - IoU_i^{pred} \right\|_2^2 \tag{3}$$

where P is the number of matched positive anchor boxes during training, IoU_i^{gt} denotes the real IoU between the $i\text{-}th$ matched detection box with corresponding ground truth box and IoU_i^{pred} indicates the predicted IoU between the $i\text{-}th$ matched detection box with corresponding ground truth box. The multi-task loss of Shifted SSD is extended to

$$L = \frac{1}{P+N} L_{conf} + \alpha \frac{1}{P} L_{loc} + \beta \frac{1}{P} L_{iou\_pred} \tag{4}$$

where N is the number of the negative anchor boxes. We find that both α and β are robust thus we empirically set both of them to 1. Figure 3 shows the structures of original SSD prediction and modified SSD prediction with IoU regression respectively.

4.3 How to ease overfitting

There are many ways to prevent overfitting in machine learning such as data augmentation, early stopping, model complexity reducing and drop out. SSD uses strong data augmentation strategy which can boost performance by 10% [25]. Here we use the same data augmentation as in SSD. Dropout [37] has been wildly used in region proposal based method like Fast RCNN and it achieves great performance in avoiding overfitting. To ease

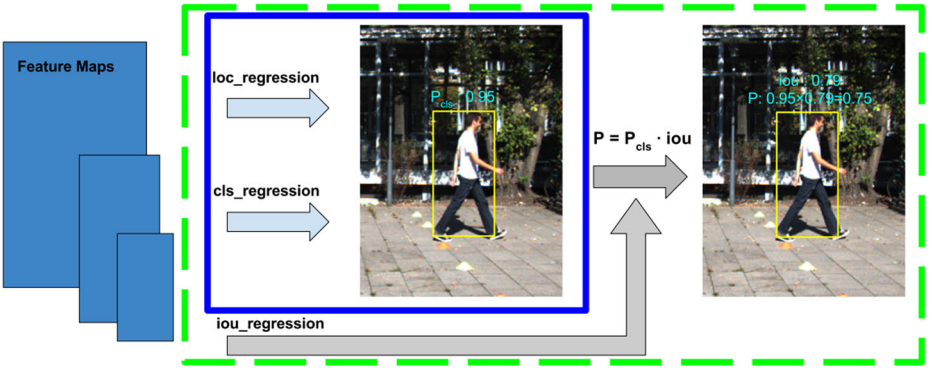


Fig. 3 Illustration of the proposed IoU-Prediction regression manner. Module inside bold blue box indicates the predictions in SSD while the Module inside bold dashed green box indicates the predictions in Shifted SSD with IoU regression

overfitting of SSD when detecting small objects, we apply channel-wise max-drop out proposed in [30] on conv4_3 before small object prediction. Max-drop layer selectively drops the maximum activations in the feature map.

4.4 Model

We adopt SSD architecture as base model and add an auxiliary shifted branch in the network to produce extra detections. Based on the previous discussion, the inputs of pool3 (stride is 2) is circularly shifted in SSD to get new feature maps and all the layers between pool3 and conv4_3 are copied to get complementary feature maps conv4_3_s for prediction. Figure 4 shows SSD model (top) and the Shifted SSD model (bottom).

The shift layer circularly shifts the input layer by s elements, where s is the input parameter. Other added layers have exactly the same settings as corresponding layers in SSD like Siamese Networks [2]. For example, the parameters of conv4_1_s convolution layer are the same as that of conv4_1. Note that the default boxes on conv4_3_s have to shift their center by half of the cell size, other parameters such as scales and aspect ratios are the same as SSD.

4.5 Model training

There are two methods to obtain the weights of the added layers, say, sharing with corresponding layers in SSD or not.

For the shared method, we just share the weights between the pre-trained SSD and our model without extra training. For example, the added conv4_3_s share the parameters with conv4_3 in SSD. It is simple and can eliminate the impact of randomness of fine tuning.

For the unshared method, we initialise the added shifted layers with the corresponding layers in the pre-trained VGG16 model and follow the same training policy as SSD. For

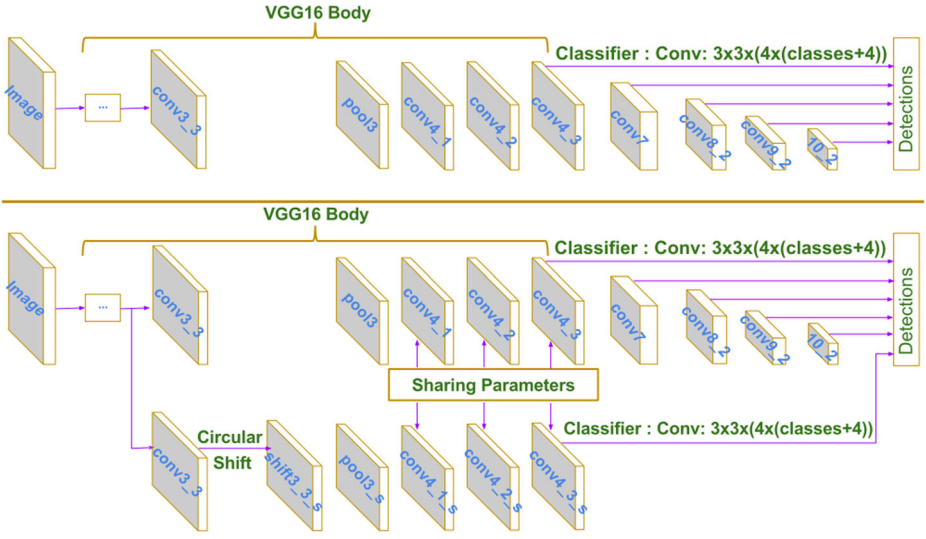


Fig. 4 Architecture of SSD (top) and Shifted SSD (bottom). A circular shift branch is added on the side of SSD. The circular shift layer circularly shift the input feature maps

anchor boxes and ground truth targets matching, we first find the most overlapped default box with each ground truth target and then any anchor boxes whose IoU with a ground truth is larger than a threshold (e.g 0.5). All the matched anchor boxes are treated as positive samples. Then we use hard negative mining strategy to select negative samples among the unmatched anchors. Last, we minimize the joint loss defined in equation (4). Table 3 compares these two methods and as can be seen, the unshared method is better.

5 Extension to sequential detection

As mentioned above, for object detection in videos, sometimes, SSD may achieve successful detection in previous frames but miss in later frames with the same detection threshold. However, many of those missed targets were detected with lower confidence than the threshold. So we use sequence information to predict the prior locations of objects to improve detection. We try methods based on trajectory hypothesis [29, 31] and KCF [16].

5.1 How to use weak detections

Detections whose confidence scores are lower than a threshold (e.g 0.3) are defined as weak detections. In order to ease the problem that every few frames SSD will miss some objects which are detected in the previous frames, we propose to use trajectory hypothesis to selectively loose the confidence threshold.

Matching strategy To generate trajectory hypothesis, we first find the nearest neighbor of each object with the biggest IoU in the previous frames like [29]. All the detections in the first frame whose detection confidence is above the threshold are treated as the start points of different trajectories. When it comes a new frame, we find the neighbors of the detections in the previous frames and this frame and push these neighbors into the corresponding trajectories. The unmatched detections in the previous frames are just duplicated into the trajectory to keep it continuous. In Fig. 5, the process of generating trajectory hypothesis is illustrated.

Poly fit predictions After obtaining a set of trajectories, poly fit method is used to predict the bounding boxes of each trajectory in new frames. Since the prior motion mode of the objects is unknown, we use linear and duplicated methods to predict the location $[x_{min}, y_{min}, x_{max}, y_{max}]$. Duplicated method means we just use the detections of the last frames as predictions.

Threshold loosing In a new frame, detections whose confidence scores are above the threshold are directly set as true positives. Then we compute the IoU between the detections whose confidence is a little bit lower than the threshold with each trajectory. The detections are set as positive and pushed to a trajectory if the IoU with that trajectory is bigger than the IoU threshold.

Table 3 Comparison between different training methods

Dataset	methods	mAP	Dataset	methods	mAP
VOC	shared	77.9	KITTI	shared	67.9
	unshared	77.9		unshared	68.0

Results on VOC 2007 and KITTI (Pedestrian) are shown

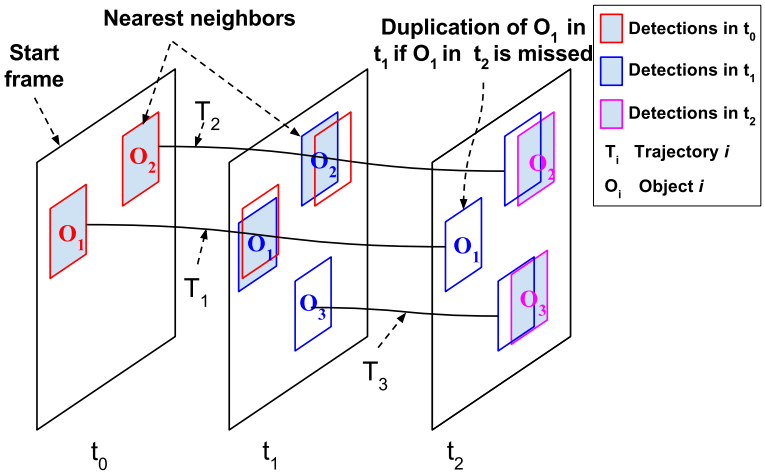


Fig. 5 Illustration of generating trajectory hypothesis

5.2 KCF prediction method

We also try KCF tracker [16] to get the predictions in a new frame. KCF trains a regressor using ground truth and regress the class of candidate windows in the new frame. For frame t, the results of frame t-1 are used as initial locations, and we train a KCF tracker for each object in frame t-1. Then these trackers are used to predict the locations of each objects in frame t. Finally, Shifted SSD is used to get detections and all the detections whose confidence is above the threshold is treated as positives. Then we loose the threshold to get the final results.

However, the bounding boxes produced by KCF tracker may deviate largely from ground truth. It will increase the number of false positives. On the other hand, though KCF is fast for single target tracking, it is relatively slow for multi-objects tracking. Assuming KCF runs at hundreds of frames-per-second and there are 10 objects in a frame, it will take 0.1 seconds to get the predictions. This is even slower than SSD.

6 Experiment results

Experiments on VOC 2007 [5], KITTI [8], MS COCO [24] and our Prisoner Monitor dataset are based on VGG16 [36], which is pre-trained on the ILSVRC CLS-LOC dataset [35]. The full testing code is built on Caffe [20] and all experiments are conducted on a Titan X (Pascal).

6.1 Pascal VOC 2007 results

Experiment settings We first train an original SSD on the VOC 2007 trainval set and VOC2012 trainval set with 300×300 input size with the same settings of [25]. After that we finetune our Shifted SSD using the unshared training method with batch size of 24 and started the learning rate at 10^{-4} for the first 20k iterations. Then learning rate decreased to 10^{-5} for another 20k iterations.

Results Analysis Table 4 compares the performance of adding shifted layers before conv7 in different directions along with using Max-drop, SNMS and denser boxes. By adding left up shifted layers we can improve mAP by 0.4% higher than SSD. We also validate that denser anchor boxes method will not help improving performance but significantly reduces the result by using smaller stride on layer pool3 and the speed is much slower.

Similar performance can be seeing with Max-drop added on conv7. We argue that it is because that PASCAL VOC 2007 dataset do not have severe overfitting problem on small objects. In the mean time, using Smooth NMS will improve mAP by 0.6%.

Finally, we add left up shifted layers before conv7 along with Smooth NMS without using max-drop and get mAP of 78.3%, 0.8% higher than SSD. Besides, the speed is almost the same with 81 fps on SSD and 77 fps on Shifted SSD on a Titan X (Pascal). Table 5 shows the comparisons with other one-stage methods. We can see that our Shifted SSD gains a relatively large improvements for classes with small objects comparing with SSD. Note that our Shifted SSD is tested with input size of 300×300 while DSSD [7] and RON [21] are tested with input size around 320×320.

We also use the detection analysis tool from [6] to compare recall with different confidences of SSD and Shifted SSD without SNMS. Figure 6 shows that Shifted SSD has larger recall than SSD and the average recall is 1.3% higher. However, mAP has little improvement on PASCAL VOC 2007. The following reasons may cause this. First, we just use the same settings as SSD300 where the smallest scale of default boxes is set relatively big in SSD for the overall results. Second, because of the data augmentation of training, SSD treats small parts of the objects as ground truth. Though Shifted SSD reduces the missing rate and improve recall, it also adds many false positives which are considered as ground truth during training and that is why recall is improved much larger than mAP. Third, as mentioned in [17], our effort to improve small objects detection may diluted by the easier cases. Figure 7 illustrates some detection results of proposed Shifted SSD on small object detection. Results of adding shifted layers before conv4_3 are similar with conv7 with slower speed.

6.2 MS COCO results

We also test Shifted SSD on MS COCO which contains more challenging small objects using shared training method without retraining due to limited time.

Table 4 VOC 2007 results

Shift Direction				Drop	SNMS	Stride	mAP
left	down	lt-up	rt-dn				
✓							77.7
	✓						77.7
		✓					**77.9**
			✓				77.8
				✓			77.6
					✓		**78.1**
						✓	70.4
	✓				✓		**78.3**
SSD300* Result							77.5

Last row means the original SSD300* [25], Drop denotes Max-drop, Stride means the stride of pool3 is 1 and column SNMS indicates Smooth NMS. The lt-up and rt-dn indicate the shift directions in left up and right down respectively

The relative best results among the listed methods are shown in bold

Table 5 Comparison with other one-stage methods on PASCAL VOC 2007

Method	mAP	aeroplane	bird	boat	bottle	cow	fps
SSD300*	77.5	79.5	76.0	69.6	50.5	81.5	**81** HZ
DeNet-101 [38]	77.1	–	–	–	–	–	33 HZ
RON320 [21]	74.2	75.7	74.8	66.1	53.2	79.5	15 HZ
DSSD321 [7]	**78.6**	**81.9**	**80.5**	68.4	**53.9**	**83.5**	10 HZ
Ours	78.3	79.2	77.4	**70.4**	51.2	82.8	77 HZ

Due to the limited space, only ap for classes with relative small size are shown. Note that our model is trained with input size of 300 × 300 which is slightly small than the input size of the others

[1]The speed of the listed methods are tested on a Titan X GPU while ours and SSD300* are tested on a Titan X (pascal) GPU

The relative best results among the listed methods are shown in bold

Experiment settings we use the pre-trained SSD300* model [25] which is trained on trainval35k [1] as base network without fine tuning. Due to limited time, we did not add max-drop and re-train the model.

Results analysis We compare the performance of adding different directions of shifted layers before conv4_3 and conv7. Table 6 shows that by adding shifted layers before conv4_3 and combine right and down shifted direction, we get mAP of 43.5%, 0.5% higher than SSD. Adding shifted layers before conv7 has the similar effect. SNMS can improve mAP significantly under more strict IoU thresholds. Finally, we combine shifted layers along with SNMS and get mAP of 43.7% and 32.8%, 0.7% and 0.9% higher than the original SSD under the IoU thresholds of 0.5 and 0.75 respectively.

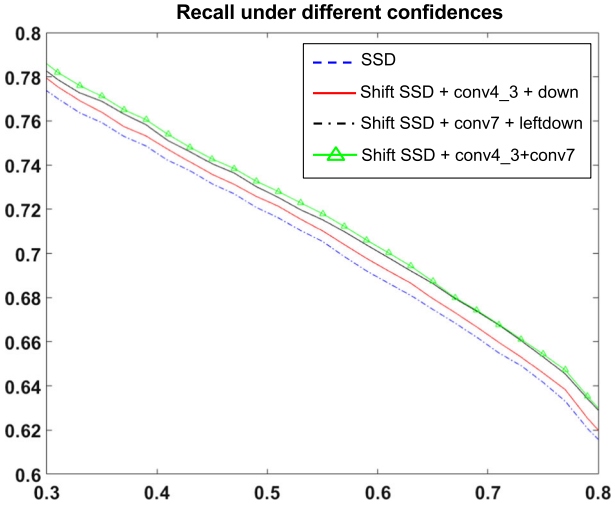


Fig. 6 Recall of SSD and Shifted SSD on VOC 2007

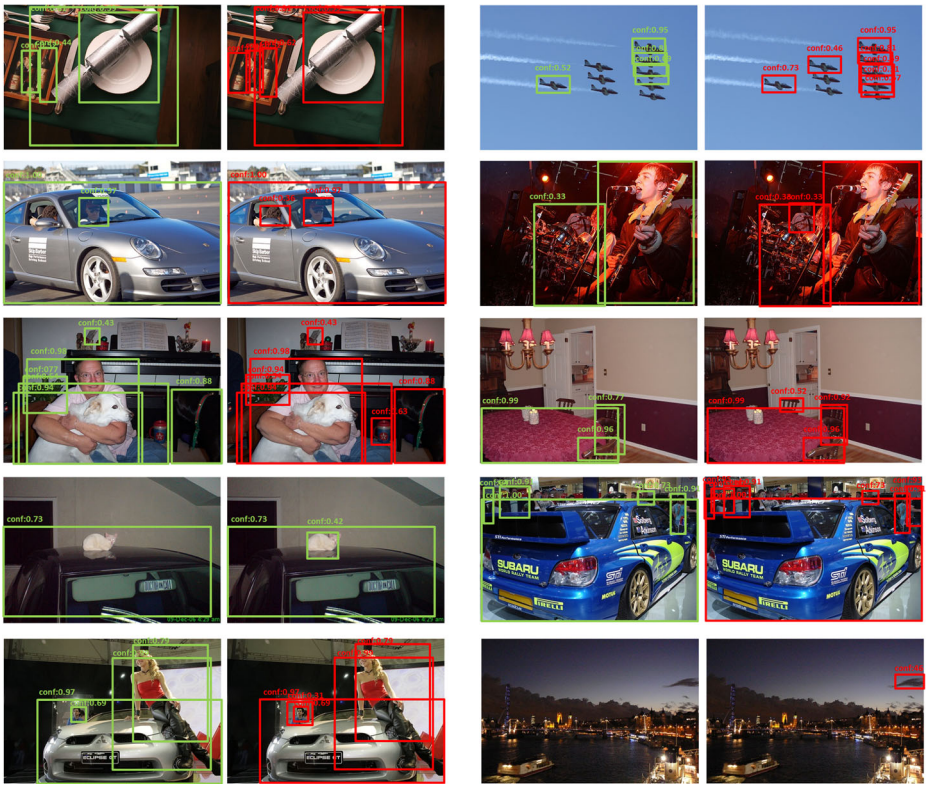


Fig. 7 Small object detection results of Shifted SSD on PASCAL VOC 2007. Image pairs are illustrated in which the left image with green bounding boxes is generated by SSD, while the right with red bounding boxes is generated by Shifted SSD. The first four rows are improved examples when using Shifted SSD and the last row is the failure examples

6.3 KITTI results

Compared with VOC 2007, KITTI contains more small objects especially for the category of pedestrian and typical image size in KITTI is about 1250 × 375.

Table 6 COCO results. Last row denotes the original SSD300* [25] result

Shift Directions				SNMS	mAP	
left	down	lt-dn	rt-up		0.5	0.75
✓					43.3	27.9
	✓				43.2	27.8
		✓			43.4	27.9
			✓		43.5	28.0
			✓	✓	**43.7**	**28.5**
SSD300* Result					43.0	27.6

The mAP under IoU thresholds of 0.5 and 0.75 is reported

The relative best results among the listed methods are shown in bold

Experiment settings We follow the protocols in [40] and split the trainval set which contains 7481 images into the training set (3682) and the validation set (3799). Our Shifted SSD only tests on pedestrian and cyclist classes. Firstly, an original SSD is trained on the training set with 1216×384 input images using conv4_3, conv7, conv6_2, conv7_2, conv8_2 and conv9_2 for prediction. For prior boxes settings, we implement k-means clustering method on the training boxes as the same procedure in [33] and set aspect ratios as $\{0.3, 0.4, 0.5, 0.7, 1.0\}$. To fit the small object detection, we set default boxes with scale $s_{min} = 0.08$ on conv4_3 and $s_{max} = 0.7$ on conv9_2. The scale of the $i\text{-}th$ layer is calculated as same as [25]:

$$s_i = s_{min} + \frac{(i-1) \times (s_{max} - s_{min})}{max - 1}, i \in [2, max - 1] \tag{5}$$

Learning rate is set to 10^{-4} for the first 30k and decreased to 10^{-5} for another 10k. After that we finetune our Shifted SSD with the batch size of 8 and initialize the learning rate as 10^{-4} for the first 10k iterations. Then learning rate is decreased to 10^{-5} for another 10k iterations. For max-drop layer, we try different drop probabilities and experimentally set it to 0.1. Lastly, we finetune the IoU-Prediction module by freezing all the weights of Shifted SSD model.

Results analysis Table 7 shows that adding the shifted layer, the Max-drop, the IoU-Prediction module and the SNMS will improve performances respectively. By adding the shifted layer, we observe a large improvement under an IoU threshold of 0.7, demonstrating that adding shifted branch will obtain more accurate localization results. Smooth NMS will significantly improve performance by 1.4% under an IoU threshold of 0.7. As shown in Table 2, KITTI dataset has a severe overfitting problem on small object detection and the result that adding max-drop on conv4_3 improves performance confirms our observation again.

Finally we combine shifted layers, max-drop and SNMS, obtaining a mAP of 70.8%, which is 3.3% higher than the original SSD under an IoU threshold of 0.5. In addition, Shifted SSD obtain a mAP of 46.1% under an IoU threshold of 0.7, which is 5.6% higher than the original SSD. Speed drops slightly from 34 fps to 31 fps. Comparing with Sub-CNN [41], our shifted SSD outperforms it by 0.8% with single input size and is faster for dozens of times.

6.4 Prison monitor results

The Prison Monitor dataset is a set of videos collected from a prison. It contains several sequences in different scenes. The task is to monitor all the prisoners in each sequence and

Table 7 KITTI results. The shown mAP is under class "Pedestrian"

Last row indicates the Sub-CNN result, Drop denotes Max-drop, column SNMS indicates Smooth NMS and IoU means IoU-Prediction. The lt-dn and rt-up indicate the shift directions in left down and right up respectively. The mAP under IoU thresholds of 0.5 and 0.7 is reported

The relative best results among the listed methods are shown in bold

Shift Directions				Drop	SNMS	IoU	mAP	
right	down	lt-dn	rt-up				0.5	0.7
✓							67.4	41.4
	✓						68.0	41.9
		✓					67.7	41.4
			✓				67.5	41.5
			✓	✓			69.4	44.7
			✓	✓	✓		70.6	46.1
			✓	✓	✓	✓	**70.8**	**47.0**
SSD 1216x38							67.5	40.5
Sub-CNN [41]							70.0	–

Table 8 Influence of adding shifted layers from different feature maps and prediction methods of trajectory hypothesis

Adding shifted layers from		Fitting methods			Recall	Precision
conv4_3	fc7	linear	duplicate	KCF		
✓					95.41	**99.62**
	✓				96.69	99.59
✓	✓				96.80	99.58
		✓			96.87	99.61
			✓		96.62	99.59
				✓	96.02	99.50
	✓	✓			**97.59**	99.61
SSD Result					95.24	99.59

The relative best results among the listed methods are shown in bold

make sure there is no missing detection. To reduce the impact of occlusion, we annotate prisoners head and shoulder as a target. All the frames have the same size at 960 × 540. In the following part of this paper, we call this dataset as PM. We randomly sample 900 frames in different sequences for training and test on the remaining ones (4400 frames).

Experiment settings We first train an original SSD500 model with 500 × 500 inputs. Like SSD, we choose conv4_ 3, conv7, conv6_2, conv7_2, conv8_2, conv9_2 and conv10_2 to predict both locations and confidences. In PM, there are many small objects whose size are around 20×20. Note that the input image needs to be resized into a fixed size 500×500, so the real size of the smallest objects are actually around 10×20. So we set default boxes with scale $s_{min} = 0.04$ on conv4_3 and $s_{max} = 0.4$ on conv9_2. We use batch size of 8 and stated the learning rate at 10^{-3} for the first 10k iterations. We then decreased it to 10^{-4} for another 10k iterations. Last we finetune the Shifted SSD model shown in the bottom part of Fig. 4 using the pretrained SSD500 model.

Results analysis Table 8 shows that by adding shifted layers before conv7 layer will significantly improve recall while the precision is almost the same. However, adding shifted layers before conv4_3 may have little effect and we believe it is because conv4_3 is effective only for the smallest objects which will not have large impact on recall on PM dataset.

Then trajectory hypothesis is added to help with detection in video sequences. Table 8 shows that using linear poly fit method is better than just duplicate the detections in last frame as predicted results. And KCF is the worst and slowest. Finally, by combining Shifted SSD using conv7 and trajectory hypothesis with linear poly fit method, we achieve the performance of recall 97.59%, precision 99.61% and 37 fps on a Titan X (Pascal). Its SSD counterpart gets recall of 95.24%, precision of 99.59% and 41 fps on a Titan X (Pascal).

7 Conclusion

We investigate the issues of small object detection and propose a novel Shifted SSD to solve these problems. Our algorithm circularly shifts lower layers of feature maps to get auxiliary feature maps to mitigate the influence of discreteness of anchor boxes method. Additionally, obtaining more accurate locations especially for small objects, two novel methods called

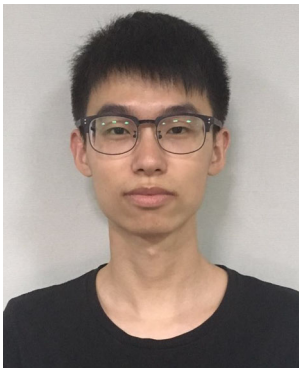
SNMS and IoU-Prediction are proposed. For video dataset, we utilize trajectory hypothesis to enhance the continuity of detection results. Our algorithm can be generalized to other deep CNN based methods that use anchor boxes mechanism to improve the performance of small object detection.

Acknowledgments This research is supported by NSFC funding (61673269, 61273285).

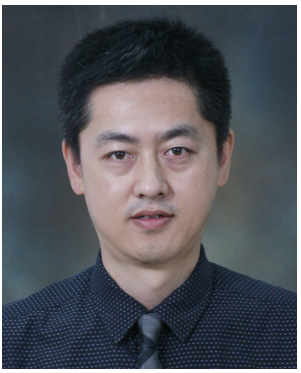
References

1. Bell S, Lawrence Zitnick C, 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
2. Bromley J, Bentz JW, Bottou L, Guyon I, LeCun Y, Moore C, Säckinger E., Shah R (1993) Signature verification using a siamese time delay neural network. Int J Pattern Recognit Artif Intell 7(04):669–688
3. Chen C, Liu MY, Tuzel O, Xiao J (2016) R-cnn for small object detection. In: Asian conference on computer vision, pp 214–230. Springer
4. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2014) Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv:1412.7062
5. Everingham M, Van Gool L, Williams C, Winn J, Zisserman A (2008) The pascal visual object classes challenge 2007 (voc 2007) results (2007)
6. Everingham M, Winn J (2007) The pascal visual object classes challenge 2007 (voc2007) development kit. University of Leeds, Tech. Rep
7. Fu CY, Liu W, Ranga A, Tyagi A, Berg AC (2017) Dssd: Deconvolutional single shot detector. arXiv:1701.06659
8. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on computer vision and pattern recognition (CVPR)
9. 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
10. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
11. 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
12. Hariharan B, Arbeláez P., Girshick R, Malik J (2015) Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 447–456
13. He K, Zhang X, Ren S, Sun J (2014) Spatial pyramid pooling in deep convolutional networks for visual recognition. In: European Conference on Computer Vision, pp 346–361. Springer
14. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. arXiv:1512.03385
15. 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
16. Henriques JF, Caseiro R, Martins P, Batista J (2015) High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596
17. Hoiem D, Chodpathumwan Y, Dai Q (2012) Diagnosing error in object detectors. In: European conference on computer vision, pp 340–353. Springer
18. Hong C, Yu J, Tao D, Wang M (2015) Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval. IEEE Trans Ind Electron 62(6):3742–3751
19. Hong C, Yu J, Wan J, Tao D, Wang M (2015) Multimodal deep autoencoder for human pose recovery. IEEE Trans Image Process 24(12):5659–5670
20. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia, pp 675–678. ACM
21. Kong T, Sun F, Yao A, Liu H, Lu M, Chen Y (2017) Ron: Reverse connection with objectness prior networks for object detection. arXiv:1707.01691
22. Li Z, Liu J, Tang J, Lu H (2015) Robust structured subspace learning for data representation. IEEE Trans Pattern Anal Mach Intell 37(10):2085–2098
23. Li Z, Liu J, Yang Y, Zhou X, Lu H (2014) Clustering-guided sparse structural learning for unsupervised feature selection. IEEE Trans Knowl Data Eng 26(9):2138–2150
24. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P., Zitnick CL (2014) Microsoft coco: Common objects in context. In: European conference on computer vision, pp 740–755. Springer

25. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: Single shot multibox detector. In: European conference on computer vision, pp 21–37. Springer
26. Liu W, Rabinovich A, Berg AC (2015). arXiv:1506.04579
27. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
28. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
29. Milan A, Leal-Taix L, Schindler K, Reid I (2015) Joint tracking and segmentation of multiple targets cvpr
30. Park S, Kwak N Analysis on the dropout effect in convolutional neural networks
31. Pirsiavash H, Ramanan D, Fowlkes CC (2011) Globally-optimal greedy algorithms for tracking a variable number of objects. In: 2011 IEEE conference on computer vision and pattern recognition (CVPR), pp 1201–1208. IEEE
32. Redmon J, Divvala S, Girshick R, Farhadi A (2015) You only look once: Unified, real-time object detection. arXiv:1506.02640
33. Redmon J, Farhadi A (2016) Yolo9000: Better, faster, stronger. arXiv:1612.08242
34. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
35. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
36. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
37. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
38. Tychsen-Smith L, Petersson L (2017) Denet: Scalable real-time object detection with directed sparse sampling. arXiv:1703.10295
39. Wang X, Han TX, Yan S (2009) An hog-lbp human detector with partial occlusion handling. In: 2009 IEEE 12th international conference on computer vision, pp 32–39. IEEE
40. Xiang Y, Choi W, Lin Y, Savarese S (2015) Data-driven 3d voxel patterns for object category recognition. In: Proceedings of the IEEE international conference on computer vision and pattern recognition
41. Xiang Y, Choi W, Lin Y, Savarese S (2017) Subcategory-aware convolutional neural networks for object proposals and detection. In: 2017 IEEE winter conference on applications of computer vision (WACV), pp 924–933. IEEE
42. Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions. arXiv:1511.07122
43. Yu J, Hong C, Rui Y, Tao D (2017) Multi-task autoencoder model for recovering human poses. IEEE Transactions on Industrial Electronics
44. Yu J, Zhang B, Kuang Z, Lin D, Fan J (2017) Iprivacy: image privacy protection by identifying sensitive objects via deep multi-task learning. IEEE Trans Inf Forensics Secur 12(5):1005–1016
45. Zhang L, Lin L, Liang X, He K (2016) Is faster r-cnn doing well for pedestrian detection?. In: European conference on computer vision, pp 443–457. Springer
46. Zhou H, Li Z, Ning C, Tang J (2017) Cad: Scale invariant framework for real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 760–768



Liangji Fang currently is working toward the Master degree at the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. His research interests include visual analysis of human, machine learning and artificial intelligence.



Xu Zhao is currently an Associate Professor in the department of Automation at Shanghai Jiao Tong University. He received the Ph.D. degree in pattern recognition and intelligence system from Shanghai Jiao Tong University in 2011. He was a visiting scholar at the Beckman Institute for Advanced Science and Technology at University of Illinois at Urbana-Champaign from 2007 to 2008. He had been the postdoctoral research fellow in the Northeastern University from 2012 to 2013. His research interests include visual analysis of human motion, machine learning and image video processing.

Shiquan Zhang currently is working toward the Master degree at the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. His research interests include visual analysis of human, machine learning and artificial intelligence.