Deep Convolutional Neural Networks for Pedestrian Detection with Skip Pooling

Jie Liu, Xingkun Gao, Nianyuan Bao, Jie Tang, and Gangshan Wu
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
MG1533026@smail.nju.edu.cn, tangjie@nju.edu.cn

***Abstract*—With the big success of deep convolutional neural networks (CNN) in image classification task, many proposal based networks are proposed to detect given objects in an image. Faster R-CNN is such a network that uses a region proposal network (RPN) to generate nearly cost-free region proposals, which has shown excellent performance in ILSVRC and MS COCO datasets. However, Faster R-CNN does not behave so well for the task of pedestrian detection since the images in popular pedestrian detection datasets have more complicated background and contain a lot of small foreground objects. In this work, we leverage the RPN architecture of Faster R-CNN and extend it to a multi-layer version combined with skip pooling to tackle the pedestrian detection problem. Skip pooling is a kind of network connection that combines multiple ROI pooling results from lower layers to form a single input to a higher layer while bypassing intermediate layers. We comprehensively evaluate our network, referred to as SP-CNN, on the Caltech pedestrian detection benchmark and KITTI object detection benchmark. Our method achieves state-of-the-art accuracy on Caltech dataset and presents a comparable result on KITTI dataset while maintaining a good speed.**

I. Introduction

Pedestrian detection has been exhaustively explored in recent years because of its growing importance in realistic applications, including automatic driving, road scene understanding or intelligent surveillance. Despite the extensive research on pedestrian detection, recent papers still show significant improvements, suggesting that we have a long way to go before reaching a saturation point.

Over the past years, a wide variety of methods have been applied to pedestrian detection. After the success of integral channel feature (ICF) detector [1], many variants [2], [3], [4], [5], [6] were proposed and showed excellent performance. A recent review of pedestrian detection [7] concludes that improved features have been driving performance and are likely to continue doing so. Recently, there has been interest in detectors derived from deep convolutional neural networks. Driven by the success of R-CNN [8] for general object detection, a series of methods adopt a two-stage pipeline for pedestrian detection. TA-CNN [9] employs the ACF detector [10] to generate proposals, and trains a R-CNN like network to jointly optimize pedestrian detection with semantic tasks; the DeepParts [11] method applies the LDCF detector [12] to generate proposals and learns a set of complementary parts by neural networks.

Despite of these hybrid pedestrian detectors, [13] uses a cascaded deep neural network to achieve real-time pedestrian detection. MS-CNN [14] proposed a multi-scale object proposal network with satisfactory detection accuracy and speed. The MS-CNN consists of a Region Proposal Network (RPN) established on multiple output layers. This architecture leverages the property that receptive fields of different layers have different scales thus providing more accurate proposals for subsequent detector and classifier. But as noted in [15], with 100 proposals per image, the RPN can achieve more than 95% recall at an IOU of 0.7, which means that there has limited space for us to further optimize the proposal quality. In contrast to the region proposal sub-network, downstream detection and classification network suffers more from the mismatch between receptive field and object size. As a result, [15] replaced the downstream detection network with boosted forests but got limited improvement on more complicated dataset like KITTI [16].

In this paper, we propose to combine faster R-CNN with skip pooling for pedestrian detection. Our work is also inspired by the work of [17] which aims at detecting small objects, but we remove the recurrent neural networks and adjust the downstream detection network to make it adapted for the task of pedestrian detection. We use VGG16 [18] net as the baseline of our network. The object detection network pools ROI from conv3, conv4 and conv5 of VGG16 simultaneously, the pooled features are then normalized, concatenated and scaled to form a fixed-length descriptor. We extensively explore the design space of our network on the popular Caltech and KITTI validation sets and evaluate it using the test sets. The evaluation results show that our network achieves the state-of-the-art performance on both Caltech and KITTI datasets.

II. Related Work

In the previous section, we have introduced many traditional methods for pedestrian detection. In this section we refer to some CNN based methods and briefly analyze their relationships with our network.

A. Faster R-CNN

Many advanced object detection networks depend on region proposal algorithms to provide object location candidates such as R-CNN, Fast R-CNN [19] and SPPnet [20], where region proposal computation has been exposed as a bottleneck. Faster R-CNN [21] introduces a Region Proposal Network (RPN) that shares convolutional features with the object detection

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

network thus eliminating almost all the region proposal computation cost. An RPN is a fully convolutional network which simultaneously predicts object bounding boxes and objectness scores at each location. The Faster R-CNN is a combination of RPN and Fast R-CNN. Our network expands the RPN architecture of Faster R-CNN to a multi-scale version.

B. RPN+BF

In [15], they argue that the original Faster R-CNN is not suited for pedestrian detection due to the downstream object detection network which degrades the detection results. They attribute this results to the insufficient resolution of feature maps for handling small objects and the lack of any bootstrapping strategy for mining hard negative examples. As a result, they propose to use an RPN followed by boosted forests on shared high-resolution feature maps. In this work, we found that the RPN+BF [15] design has limited improvement on KITTI dataset and our method outperforms RPN+BF by around 15% using the metric of average precision in moderate mode.

C. MS-CNN

The MS-CNN consists of a proposal sub-network and a detection sub-network. The proposal sub-network is a multi-scale version of the RPN in Faster R-CNN, where detection is performed at multiple output layers thus receptive fields match objects of different scales. This multi-scale RPN can provide better proposals for small objects than that of Faster R-CNN so that they made a big improvement on the KITTI dataset which contains many small objects. Our work shows that the detection sub-network is a bottleneck of MS-CNN and we propose a new detection sub-network which further improves the pedestrian detection results on KITTI dataset.

III. Architecture

In this section, we introduce SP-CNN, a detector with skip ROI pooling for pedestrian detection.

A. Complementary region proposal networks

Inspired by MS-CNN, as shown in Fig. 1, we decide to enhance the region proposal network by establishing an independent RPN on different layers respectively such that each RPN can only be responsible for training samples with a given scale. During training, the weights $\mathbf{W}$ of the RPN are learned from a set of training samples $S = \{(I_i, y_i, B_i)\}_{i=1}^{N}$. I_i is a training image patch. y_i is the class label of I_i, background class always has a label of 0. Vector $B_i = (b_i^x, b_i^y, b_i^w, b_i^h)$ is the bounding box coordinates, where b_i^w and b_i^h is the width and height of the bounding box. N corresponds to the number of training patches. We use a multi-task loss L on each image patch to jointly train for region proposal classification and bounding box regression:

$$L(\mathbf{W}) = \sum_{m=1}^{M} \sum_{i \in S^m} \alpha_m l^m(I_i, y_i, B_i|\mathbf{W}) \qquad (1)$$

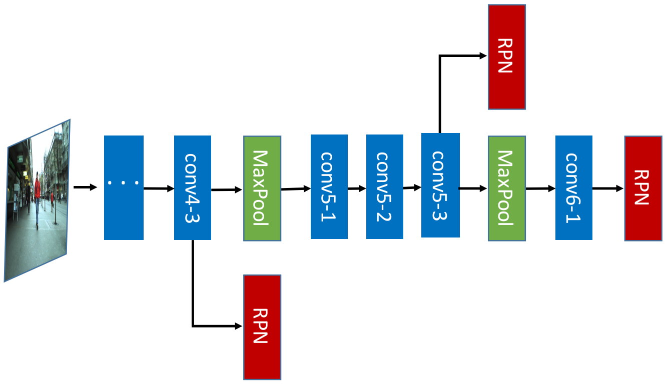


Fig. 1: The multi-scale region proposal networks. The baseline network is VGG16 and the three RPNs are established on conv4-1, conv5-3 and conv6-1 respectively. They work jointly to provide region proposals for downstream pedestrian detector. More details can be found in [14].

where M is the number of proposal sub-networks, α_m is the weight of loss l^m, and $S = \{S^1, S^2, \ldots, S^M\}$ with S^m the set of training examples of scale m and it's the only contribution to the loss of proposal network m. Inspired by the success of joint learning of classification and bounding box regression [19], [21], the loss of each proposal network combines these two objectives:

$$l(I, y, B|\mathbf{W}) = L_{cls}(p(I), y) + \lambda[y \geq 1] L_{loc}(b, \hat{b}) \qquad (2)$$

Here, $p(I) = (p_0(I), p_1(I), \ldots, p_C(I))$ is the probability distribution over classes and $L_{cls}(p(I), y) = -\log p_y(I)$ is the cross entropy loss. λ is a trade-off coefficient between classification loss and regression loss. $\hat{b} = (\hat{b}_x, \hat{b}_y, \hat{b}_w, \hat{b}_h)$ is the regressed bounding box, and

$$L_{loc}(b, \hat{b}) = \frac{1}{4} \sum_{j \in \{x,y,w,h\}} smooth_{L1}(b_j, \hat{b}_j) \qquad (3)$$

the smoothed bounding box regression loss of [19]. The term $y \geq 1$ means the bounding box loss is activated only for positive samples since background samples hold $y = 0$. The optimal parameters $\mathbf{W}^\star = \arg\min_W L(\mathbf{W})$ are learned by stochastic gradient descent.

B. Skip-layer Pooling

It's a classic idea to connect layers with skip paradigm in neural networks, where activations from a lower layer are routed directly to a higher layer while bypassing intermediate layers. The specifics of the wiring and combination method differ between models and applications. Our usage of skip connections is most closely related to those used by Sermanet et al. [22]. In [22], they directly combine activations from

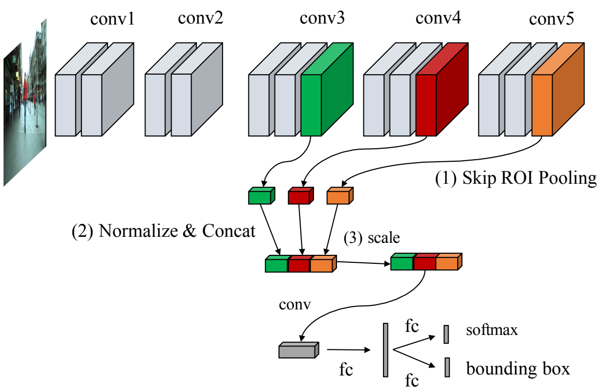


Fig. 2: Object detection sub-network with skip pooling. We pool from conv3, conv4 and conv5 simultaneously and the activations are L2 normalized, concatenated and scaled to form a fixed-length feature descriptor.

different layers together, but in section V-A we argue that it is essential to L2 normalize the activations prior to combining them. The activation normalization trick when combining features across layers was recently noted by ParseNet [23] in a model for semantic segmentation that makes use of global image context features. Skip connections have also been applied in recent models for semantic segmentation, such as the "fully convolutional networks" in [24], and for object instance segmentation, such as the "hypercolumn features" in [25].

The skip pooling architecture of our detection network is depicted in Fig. 2. Different from prior networks such as Fast R-CNN, Faster R-CNN, and SPPnet, which pool from the last convolutional layer ("conv5-3") in VGG16, we pool from multiple layers ("conv3-3, conv4-3, conv5-3") for each ROI. These pooled features are then L2-normalized, concatenated and rescaled to produce a fixed-length feature descriptor. In [17], they feed the fixed-length feature descriptor into a 1x1 convolution layer to produce a matched shape that previously trained VGG16 network holds. They preserve the existing layer shapes to benefit from pre-training, but in turn, it limits the design space of downstream detector's architecture. We propose a new detector that has different layers and dimensions from VGG16, thus there is no need to keep the layer shape compatible with VGG16. As a result, we modify the filter of the convolutional layer, following scale layer, to be of size 3x3. To compensate the loss of amplitudes after L2 normalization, our scale layer uses a fixed scale of 1000.

C. Sampling

We adopt the bootstrapping sampling strategy described in [14] to generate the training set $S = \{P, N\}$ for each proposal detector, where P is the positive samples and N the negative samples. A bounding box B is considered as positive if its IoU (Intersection over Union) B_{iou} is beyond 0.5, where

$$B_{iou} = \max_{i \in Q} IoU(B, B_i^\star) \tag{4}$$

Q is the set of ground truth and IoU the intersection over union between anchor B and ground truth bounding box $B_i^\star$. The corresponding class label and ground truth bounding box of anchor B is y_j and $B_j^\star$,where $j = \arg\max_{j \in Q} IoU(B, B_j^\star)$ and (I, y_j, B) is added to the positive set P. All the positive samples in P with $y \geq 1$ contribute to the loss. Samples with IoU less than 0.2 are considered as negative training sample candidates and the remaining samples are discarded. With the strategy of bootstrapping, the final negative training samples N are γ times the size of P.

IV. Experiments

A. Experimental setup

We train and evaluate our model on two popular datasets: Caltech [26] and KITTI [16]. We use caffe [27] framework to train the multi-scale region proposal network and the pedestrian detection network. All the experiments use a pre-trained VGG16 model downloaded from the Caffe model zoo. The training process consists of two stages. The first stage uses random sampling and a small trade-off coefficient λ to train the RPN. We train the fist stage for 10k iterations with a base learning rate of 0.00005. The second stage is initialized using the weights trained in the first stage. In the second stage bootstrapping sampling is used and we set $\lambda = 1$.

name	hr	vr	ar	overlap	filter
All	[20 inf]	[0.2 inf]	0	0.5	1.25
Reasonable	[50 inf]	[0.65 inf]	0	0.5	1.25
Scale=near	[80 inf]	[inf inf]	0	0.5	1.25
Scale=medium	[30 80]	[inf inf]	0	0.5	1.25
Scale=large	[100 inf]	[inf inf]	0	0.5	1.25
Ar=all	[50 inf]	[inf inf]	0	0.5	1.25
Ar=typical	[50 inf]	[inf inf]	0	0.5	1.25
Occ=partial	[50 inf]	[0.65 1]	0	0.5	1.25
Occ=heavy	[50 inf]	[0.2 0.65]	0	0.5	1.25
Overlap=25	[50 inf]	[0.65 inf]	0	0.25	1.25
Overlap=50	[50 inf]	[0.65 inf]	0	0.5	1.25
Expand=125	[50 inf]	[0.65 inf]	0	0.5	1.25
Expand=150	[50 inf]	[0.65 inf]	0	0.5	1.50

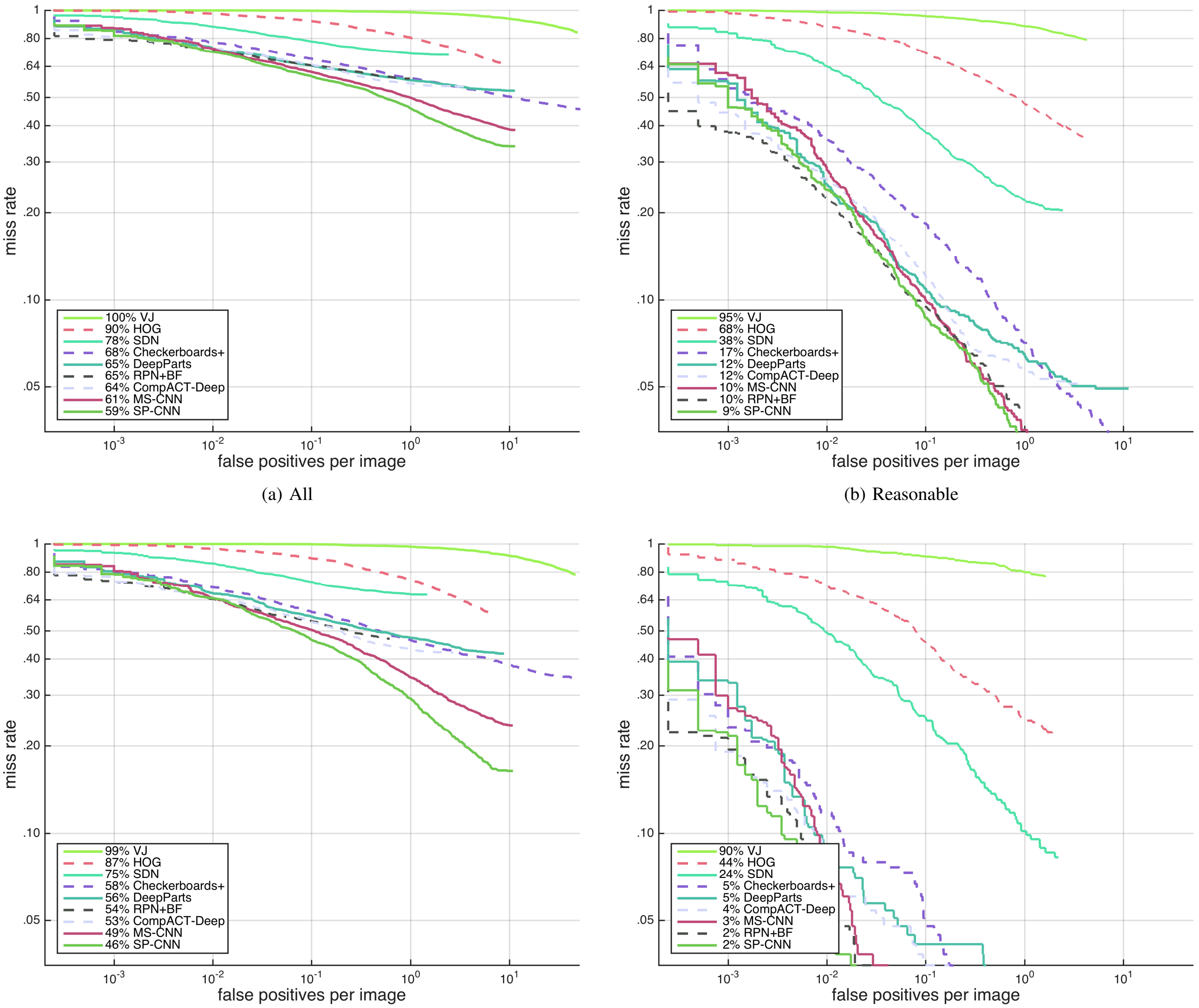
TABLE I: Testing Scenarios (part). **name**: experiment name, **hr**: height range to test, **vr**: visibility range to test, **ar**: aspect ratio range to test, **overlap**: overlap threshold for evaluation, **filter**: expanded filtering.

B. Caltech Pedestrian Dataset

The Caltech Pedestrian Dataset consists of approximately 10 hours of 640x480 30Hz video taken from a vehicle driving through regular traffic in an urban environment. About 250,000 frames (in 137 approximately minute long segments) with a total of 350,000 bounding boxes and 2,300 unique pedestrians

method	All	Reasonable	Scale=near	Scale=medium	Scale=large	Ar=all	Ar=typical	Occ=partial	Occ=heavy	Overlap=25	Overlap=50	Expand=125	Expand=150
VJ	99.5	94.7	89.9	99.4	86.2	94.2	93.8	98.7	98.8	91.3	94.7	94.7	96.1
HOG	90.4	68.5	44.0	87.4	37.9	66.5	62.8	84.5	96.0	66.6	68.5	68.5	71.9
SDN	78.4	37.9	23.7	74.6	18.5	36.5	34.1	49.4	78.8	33.8	37.9	37.9	37.9
Checkerboards+	67.7	17.1	4.9	58.0	2.4	15.1	13.4	31.3	77.9	8.9	17.1	17.1	17.1
DeepParts	64.8	11.9	4.8	56.4	4.4	10.6	8.7	19.9	60.4	13.3	11.9	11.9	11.9
CompACT-Deep	64.4	11.7	4.0	53.2	2.6	9.6	7.0	25.1	65.8	9.1	11.7	11.7	11.7
RPN+BF	64.7	9.6	2.3	53.9	**1.2**	7.7	6.0	24.2	74.4	7.9	9.6	9.6	9.6
MS-CNN	60.9	10.0	2.6	49.1	2.0	8.2	6.3	**19.2**	60.0	7.1	10.0	10.0	10.0
SP-CNN(ours)	**58.6**	**9.1**	**1.8**	**45.6**	1.6	**7.2**	**5.5**	21.5	**58.0**	**6.8**	**9.1**	**9.1**	**9.1**

TABLE II: Caltech evaluation results (more details). This table shows the results of more evaluation scenarios, our method achieves state-of-the-art accuracy in most of the scenarios thus is more stable than RPN+BF and MS-CNN. The metric is miss rate at $fppi = 10^{-1}$.



(a) All

(b) Reasonable

(c) Scale=medium

(d) Scale=near

Fig. 3: Evaluation results of “All”, “Reasonable”, “Scale=medium” and “Scale=near” scenarios on the Caltech Pedestrian Dataset. Our approach is referred to as SP-CNN. The legend is ranked according to the miss rate at $fppi = 10^{-1}$. Details of these scenarios are shown in TABLE I.

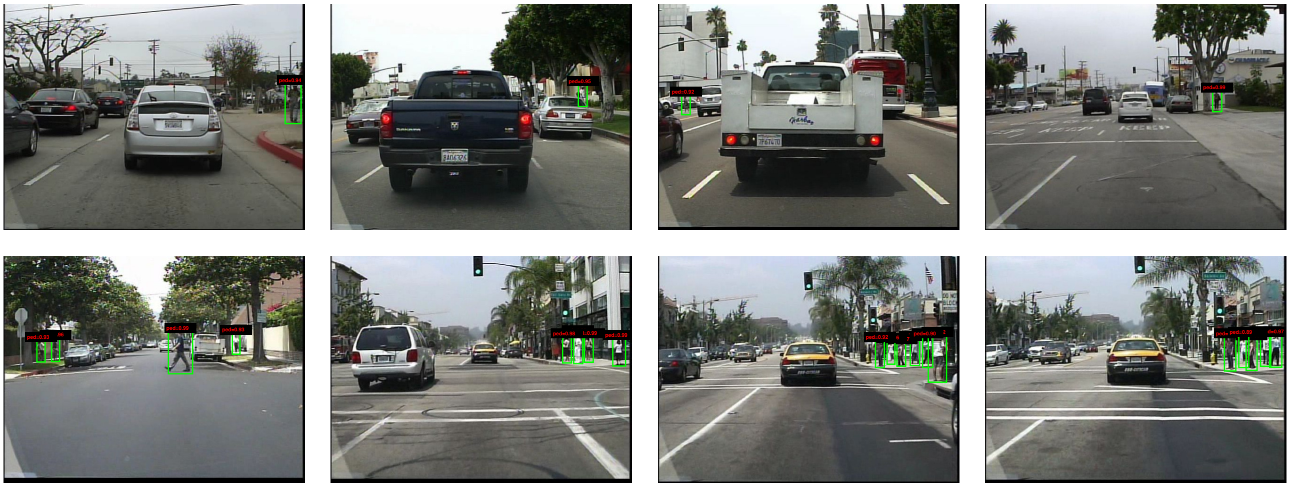


Fig. 4: Selected examples of pedestrian detection results on the Caltech test set using our network. The training set is set01-set05 and the test set is set06-set10. Our network can detect pedestrians with small heights very well. Each bounding box is associated with a softmax score in [0, 1]. A score threshold of 0.8 is used to display these images.

were annotated. The annotation includes temporal correspondence between bounding boxes and detailed occlusion labels. The training data (set00-set05) consists of six training sets, each with 6-13 one-minute long sequence files, along with all annotation information. The testing data (set06-set10) consists of five sets and evaluation is performed every 30th video frame.

We evaluate the results using the Matlab code provided with the dataset and compare with other methods including Checkerboards+ [28], CompACT-Deep [29], DeepParts [30], MS-CNN [14], and RPN+BF [15]. The metric used in evaluation is log-average Miss Rate on False Positive Per Image (FPPI) in $[10^{-2}, 10^{0}]$. The most relevant working point is defined as position where FPPI = 10^{-1}, which means, only one false detection every ten images is allowed. The miss rate for each of the detectors at this working point is shown in the legend, more details can be found in [26]. Fig. 3 shows a part of the evaluation results. They correspond to 4 testing scenarios which are "All", "Reasonable", "Scale=near", and "Scale=medium" respectively. Details of the 4 testing scenarios are shown in TABLE I and a more detailed results including most of the testing scenarios are shown in TABLE II. In the "All" scenario, our method has an MR of 59% ,which is 2 percentage points lower than MS-CNN and 6 points lower than RPN+BF. In the "Reasonable" scenario, our method has a miss rate of 9% which is 1 point lower than MS-CNN and RPN+BF after rounding. The "All" scenario has more small pedestrian objects than "Reasonable" as its height range is from 20 to infinity while "Reasonable" is from 50 to infinity. Although the visibility range is different between "All" and "Reasonable", further experiments show that with the same visibility range as "All", our method still has an MR of 9% in the "Reasonable" scenario. So we can get a conclusion that our method is more friendly to small objects thus outperforms the other methods in the "All" scenario. In the "scale=medium" scenario, our method outperforms the closest competitor by 3 points while in the "scale=near" scenario, the miss rate of our method is slightly better than RPN+BF and MS-CNN, which gives a further evidence that our detector misses fewer small objects. Table III showes the running time on Caltech. Our method has a comparable test speed while achieving a lower miss rate than RPN+BF. Selected examples of pedestrian detection results on the Caltech test set are shown in Fig. 4.

method	hardware	time/img(s)	MR (%)
LDCF[]	CPU	0.6	25
CCF[]	Titan Z GPU	13	17
CompACT-Deep[]	Tesla K40 GPU	0.5	12
RPN+BF[]	Tesla K40 GPU	0.5	10
SP-CNN[ours]	Tesla K20 GPU	**0.36**	**9**

TABLE III: Comparisons of running time on the Caltech dataset.

C. KITTI

The KITTI dataset consists of video frames from autonomous driving scenes, with 7,481 images for training and 7,518 images for testing. Since the ground truth annotations of the KITTI test set are not released, we use two strategies to split the KITTI training set into a train set and a validation set. MS-CNN and SubCNN [31] are the top two methods on KITTI dataset in published works so far. To compare with SubCNN, we split the training set into 3,682 training images and 3,799 testing images. To compare with MS-CNN. we split the training set into 3,712 training images and 3,769 testing images. The metric we use is average precision and there are three evaluation modes which are shown in TABLE IV. The validation results are shown in TABLE V and TABLE VI.

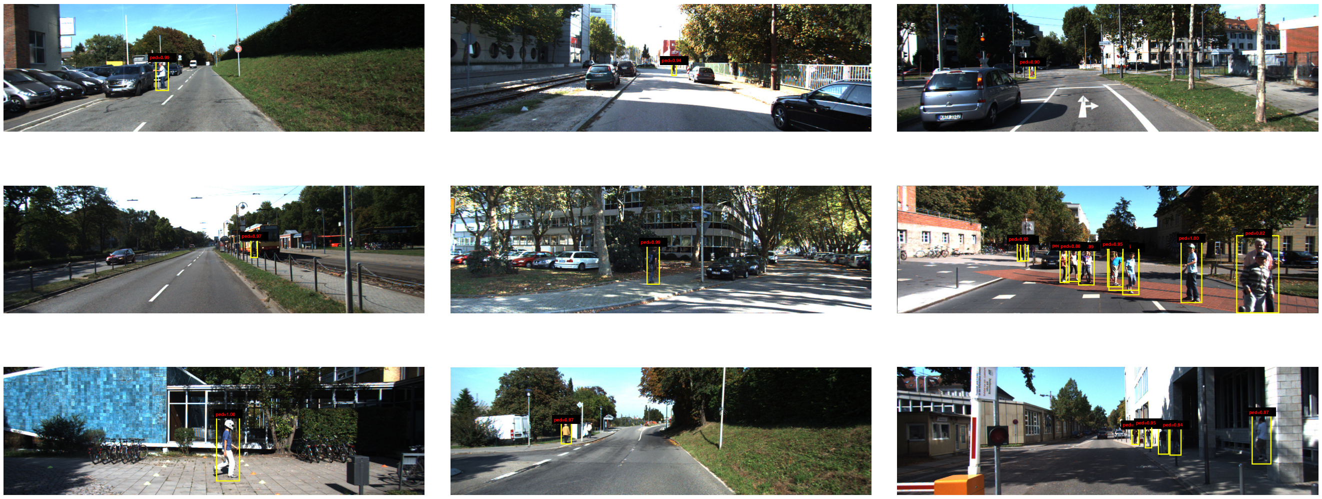


Fig. 5: Selected examples of pedestrian detection results on the KITTI test set using our network. The training set contains 3,712 images and the validation set contains 3,769 images. Our network can detect pedestrians with small heights very well. Each bounding box is associated with a softmax score in [0, 1]. A score threshold of 0.8 is used to display these images.

The moderate test set contains images with a minimum height of 25 pixels indicating that there may be many small objects to detect. Our method leverages skip pooling methodology to improve the detection accuracy thus achieving better results than SubCNN and MS-CNN in moderate mode. The other two modes are also got improved by using our network. Selected examples of pedestrian detection results on the KITTI dataset are shown in Fig. 5

mode	Min. height (Px)	Max. occlusion	Max. truncation (%)
easy	40	Fully visible	15
hard	25	Difficult to see	50
moderate	25	Partly occluded	30

TABLE IV: KITTI evaluation modes. Different modes correspond to different difficulties and all methods evaluated on the benchmark server are ranked based on the moderately difficult results.

method	easy	moderate	hard
SubCNN	86.43	69.95	64.03
Ours	**90.70**	**86.55**	**78.52**

TABLE V: Comparison between SubCNN and Our network on the KITTI validation set. Our method outperforms SubCNN in all three modes.

method	easy	moderate	hard
MS-CNN	76.38	72.26	64.08
Ours	**82.40**	**77.06**	**69.16**

TABLE VI: Comparison between MS-CNN and Our network on the KITTI validation set. MS-CNN is the previous state-of-the-art method on KITTI dataset.All methods submitted to KITTI benchmark server are ranked based on the moderately difficult results, our method improves the mAP from 72.26% to 77.06% in moderate mode.

V. Design evaluation

A. Which layers to pool from?

Our network uses skip pooling strategy which pools regions of interest (ROI) from multiple layers and then normalize, concatenate and re-scale these features. There are several convolutional layers in VGG16, the problem is that we should pool from which layers. A straightforward approach is to pool the ROI from each layer and then use a convolutional layer to reduce the dimensionality. However, this may not work as illustrated later. To get a better insight of this problem, we consider several combinations and evaluate them separately, the evaluation results are shown in TABLE VII. When pooling from conv3, conv4 and conv5 simultaneously and using L2 normalization together with a scale layer, we get a miss rate of 9.8% which is the best among all the combinations. The normalization and scaling strategy significantly improves the detection accuracy.

ROI pooling from:			**Merge features using:**	
conv3	conv4	conv5	conv	L2+scale+conv
		√	15.2%	10.0%
	√	√	10.8%	10.0%
√	√	√	12.5%	**9.1%**

TABLE VII: Combining features from different layers. Metric: miss rate at $fppi = 10^{-1}$. Training set: set01-set05 of Caltech dataset. Testing set: set06-set10 of Caltech dataset.

B. How should we set the scale factor?

In VGG16, the output features at different layers can have very different amplitudes, so that directly combine them may lead to unstable learning. It is necessary to normalize the amplitude such that the features being pooled from multiple layers have similar magnitude. To compensate for the normalization, we explicitly re-scale the features with a empirically determined factor. TABLE VIII shows the effects of using factors with different orders.when scaled by 1000, our network gets the best results on both of the Caltech and KITTI datasets.

scale factor	Caltech (MR)	KITTI(mAP)
1	49.4%	31.8%
10	22.3%	19.6%
100	10.4%	73.6%
1000	**9.1%**	**76.3%**

TABLE VIII: Evaluation results of different scales. For Caltech the metric is miss rate and for KITTI the metric is mean average precision.

Input image size	Caltech (MR)	KITTI (mAP)
384	12.5%	71.2%
576	11.3%	74.3%
768	**9.1%**	**77.1%**

TABLE IX: Evaluation results of different input image sizes. For Caltech the metric is miss rate and for KITTI the metric is mean average precision.

C. The effect of input image size

The input image size can be a critical factor for object detection with convolutional neural networks, an appropriate image size means a better detection accuracy. The selection of input image size depends on the specifics of each network structure. With input image up-sampling, we can usually get feature maps with a higher resolution which is very helpful for detection small objects. TABLE IX evaluates different input image size on Caltech and KITTI datasets.

D. Embedded with deconvolution layer

Our network is initialized with a pre-trained ImageNet classification model. This pre-trained model uses an input image size of 224x224, we can benefit most from this model if using a similar image patch size. But images in Caltech or KITTI dataset are much larger than ImageNet, there are many small objects in these images. For these small objects, the corresponding feature maps of higher convolutional layer are very weak and less discriminative. Fast R-CNN and Faster R-CNN explicitly up-sample the input images (by $\sim$2 times) to get a stronger response for small objects. Our method also up-samples the original images and get a better detection accuracy. However the up-sampling strategy does not directly improve resolution of the convolutional feature maps, a better approach is to use feature map approximation since it reduces memory usage and explicitly increases the resolution of feature maps. When using deconvolution the average precision of moderately difficult results improves from 74.3% to 76.2% on the KITTI validation set.

VI. Conclusion

In this work, we have presented a deep convolutional neural network that combines a multi-scale RPN and a novel object detection network where we leverage the skip pooling paradigm to improve the detection accuracy for both large and small objects. To get the final network structure, we have considered varieties of hyper parameters and evaluated them on Caltech and KITTI validation sets. Furthermore, we have also investigated the usage of a deconvolutional layer before ROI pooling which further improves the performance of our detector. We comprehensively evaluate our network on Caltech and KITTI datasets, our method achieves state-of-the-art performance on Caltech dataset and presents a comparable result on KITTI dataset while maintaining a good speed.

Acknowledgment

This paper is partly funded by Research Project of Excellent State Key Laboratory (61223003), Research Fund of the State Key Laboratory for Novel Software Technology at Nanjing University (ZZKT2016B09). The corresponding author of this paper is Jie Tang.

References

[1] P. Dollr, Z. Tu, P. Perona, and S. Belongie, “Integral channel features,” in *British Machine Vision Conf. (BMVC)*, 2009, pp. 91-1.

[2] S. Zhang, C. Bauckhage, and A. B. Cremers, “Informed haar-like features improve pedestrian detection,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 947-954.

[3] S. Zhang, R. Benenson, and B. Schiele, “Filtered channel features for pedestrian detection,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1751-1760.

[4] W. Nam, P. Dollr, and J. H. Han, “Local decorrelation for improved detection,” in *NIPS*, 2014.

[5] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, “Strengthening the effectiveness of pedestrian detection with spatially pooled features,” in *European Conf. on Computer Vision (ECCV)*, 2014, pp. 546-561.

[6] S. Zhang, C. Bauckhage, D. A. Klein, and A. B. Cremers, “Exploring human vision driven features for pedestrian detection,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 25, no. 10, 2015, pp. 1709-1720.

[7] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, “How far are we from solving pedestrian detection?” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1259-1267.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580-587.

[9] Y. Tian, P. Luo, X. Wang, and X. Tang, “Pedestrian detection aided by deep learning semantic tasks,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5079-5087.

[10] P. Dollar, R. Appel, S. Belongie, and P. Perona, “Fast feature pyramids for object detection,” *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 36, no. 8, 2014, pp. 1532-1545.

[11] Y. Tian, P. Luo, X. Wang, and X. Tang, “Deep learning strong parts for pedestrian detection,” in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2015, pp. 1904-1912.

[12] W. Nam, P. Dollar, and J. Hee Han, “Local decorrelation for improved pedestrian detection,” in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[13] A. Angelova, A. Krizhevsky, V. Vanhoucke, A. Ogale, D. Ferguson, “Real-Time Pedestrian Detection With Deep Network Cascades,” in *British Machine Vision Conf. (BMVC)*, 2015, pp. 32-1.

[14] Z. Cai, Q. Fan, R. Feris, and N. Vasconcelos, “A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection,” in *European Conf. on Computer Vision (ECCV)*, 2016, pp. 354-370.

[15] L. Zhang, L. Lin, X. Liang, and K. He, “Is Faster R-CNN Doing Well for Pedestrian Detection?” in *European Conf. on Computer Vision (ECCV)*, 2016, pp. 443-457.

[16] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354-3361.

[17] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, “Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2874-2883.

[18] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *ICLR*, 2015.

[19] R. Girshick, “Fast R-CNN,” in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2015, pp. 1440-1448.

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” in *European Conf. on Computer Vision (ECCV)*, 2014, pp. 346-361.

[21] S. Ren, K. He, R. Girshick, J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 91-99.

[22] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, “Pedestrian detection with unsupervised multi-stage feature learning,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3626-3633.

[23] W. Liu, A. Rabinovich, and A. C. Berg, “ParseNet: Looking wider to see better,” *arXiv e-prints, arXiv:1506.04579 [cs.CV]*, 2015.

[24] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431-3440.

[25] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, “Hyper-columns for object segmentation and fine-grained localization,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 447-456.

[26] P. Dollar, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: An evaluation of the state of the art,” *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, no. 4, 2012, pp. 743-761.

[27] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *arXiv preprint arXiv:1408.5093*, 2014.

[28] S. Zhang, R. Benenson, and B. Schiele, “Filtered channel features for pedestrian detection,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1751-1760.

[29] Z. Cai, M. Saberian, and N. Vasconcelos, “Learning Complexity-Aware Cascades for Deep Pedestrian Detection,” in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2015, pp. 3361-3369.

[30] Y. Tian, P. Luo, X. Wang, and X. Tang, “Deep Learning Strong Parts for Pedestrian Detection,” in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2015, pp. 1904-1912.

[31] Y. Xiang, W. Choi, Y. Lin, S. Savarese, “Subcategory-aware Convolutional Neural Networks for Object Proposals and Detection,” *arXiv preprint arXiv:1604.04693*, 2016.