SCAN: Semantic Context Aware Network for Accurate Small Object Detection

Linting Guan [1, 2] , **Yan Wu** [1*], **Junqiao Zhao** [1]

[1] *College of Electronics & Information Engineering, Tongji University,*
Telecom Building, 4800 Cao'an Road, Jiading,
Shanghai, 201804, China

E-mail: {glinting,yanwu, zhaojunqiao}@tongji.edu.cn

,

[2] *College of Mathematics, Physics and Information Science,*
Zhejiang Ocean University,
1 South Haida Road, Lincheng,
Zhoushan, Zhejiang 316004, China

Received July 15, 2017

Accepted March 28, 2018

Abstract

Recent deep convolutional neural network-based object detectors have shown promising performance when detecting large objects, but they are still limited in detecting small or partially occluded ones—in part because such objects convey limited information due to the small areas they occupy in images. Consequently, it is difficult for deep neural networks to extract sufficient distinguishing fine-grained features for high-level feature maps, which are crucial for the network to precisely locate small or partially occluded objects. There are two ways to alleviate this problem: the first is to use lower-level but larger feature maps to improve location accuracy and the second is to use context information to increase classification accuracy. In this paper, we combine both methods by first constructing larger and more meaningful feature maps in top-down order and concatenating them and subsequently fusing multilevel contextual information through pyramid pooling to construct context aware features. We propose a unified framework called the Semantic Context Aware Network (SCAN) to enhance object detection accuracy. SCAN is simple to implement and can be trained from end to end. We evaluate the proposed network on the KITTI challenge benchmark and present an improvement of the precision.

Keywords: Deep learning; object detection; semantic features.

1. Introduction

Object detection is a key element for a safe and robust autonomous driving system. Traditional detection methods are based on both engineered features such as histogram of oriented gradients (HOG)[6] and scale-invariant feature transform (SIFT)[22] and on explicit detection models such as deformable part model (DPM)[7] and its variants. The main idea is to design special rules for extracting descriptive information from images to enable object identification. These traditional methods have been prevalent in the past ten years and have achieved remarkable results. However, it is both difficult and time-consuming to design effective features for special vision tasks. Moreover, these features are not robust due to the variability in the shape of objects and environmental changes in illumination.

Recently, the success of modern neural networks[16] on the ImageNet classification

[*] Correspond Author.

challenge[25], deep convolutional neural networks (DCNN) have become the most promising method for vision recognition tasks. The region-based convolutional network (R-CNN)[9] is one of the most successful methods based on DCNN; it used regions acquired with off-the-shelf methods such as selective search[29], Edge Boxes[36] and multiscale combinatorial grouping (MCG)[1] and then classified the proposed regions using DCNN and regressing the bounding box of the region. Because DCNN could accurately classify the proposed regions, R-CNN dramatically improved the state-of-the-art of object detection tasks on the PASCAL VOC benchmark.

However, the task of object detection and image classification has a internal conflict. Image classification tasks must be insensitive to object translation, scaling and rotation to make the learned features less sensitive to location; however, these features are detrimental to precisely locating objects in an object detection task. This problem is exacerbated when detecting small or partially occluded objects because they have very limited information due to small areas in the image; consequently, it is difficult for the neural network to extract distinguishing features that can precisely locate and classify such objects. Acquiring the contextual information surrounding these objects is a key component in identifying them correctly because the contextual information helps to exclude inappropriate object classes and increases the probability of identifying appropriate classes by considering the background (e.g., a boat has a higher probability of appearing on the sea than does a house).

The goal of this paper is to alleviate the above two problems in a simple but effective way. We propose a novel neural network structure called the Semantic Context Aware Network (SCAN) to tackle the above problems. SCAN introduces two additional components: a Location Fusion Module (LFM) and a Context Fusion Module (CFM). The LFM constructs semantic features by making full use of internal features with appropriate sizes to obtain additional object location information, while the CFM uses multilevel context information through pyramid pooling to construct context-aware features. The network is simple to implement and can be trained end to end.

The rest of this paper is organized as follows. Section 2 describes related object detection methods. Section 3 describes the details of the proposed framework. We provide an evaluation of our method in Section 4 . Finally, conclusions are given in Section 5 .

2. Related Work

One of the most important components of object detection is computing appropriate features from images. Over the past decade, HOG and SIFT have been the most prevalent methods for designing hand-engineered features. Various versions of HOG and SIFT features combined with support vector machine (SVM) classifiers have been employed to construct numerous object detection systems. However, due to breakthrough improvements in image classification tasks, DCNN has now become a more promising feature extraction method. Object detection methods based on DCNN such as R-CNN have demonstrated impressive improvements in detection accuracy. R-CNN converted the detection problem to a classification problem by identifying regions in images using algorithms such as selective search[29]. Then, it classified the regions using DCNN. SPPnet[10] improved the detection speed by sharing feature maps extracted from entire image and using multiple scales of region-of-interest (RoI) pooling to perform region classification. Fast R-CNN[8] used a single-scale RoI pooling layer to compute equally-sized features for region classification, which improved the detection accuracy by integrating object classification losses and bounding box regression losses as unit losses for optimization. Faster R-CNN[24] further improved the detection speed by proposing candidate regions on shared feature maps called a Region Proposal Network (RPN).

Recently, some works have used middle-level features to fill gaps in object detection and recognition. Kong *et al.*[15] introduced a new way to build more expressive middle-level features by fusing low-, middle- and high-level features for object detection. Cai *et al.*[2] detected objects of different sizes by proposing multilevel regions in feature maps. Xiang *et al.*[31] enhanced the accuracy of detecting occluded or truncated objects by using subcategory related information to guide the bounding box proposal. Yang *et al.*[32] used scale-dependent pooling to improve detection accuracy and improve detection speed by rejecting easy negative-object proposals. Shrivastava, Lin *et al.*[28,20] constructed additional semantic feature maps in a top-down manner, and detected objects from multilevel feature maps.

Context carries important information when an object is partially occluded or its size is small. Context has been exploited in many traditional detection models[3,17]. Recently, some DCNN-based works have used context information to enhance object detection accuracy. For example, Vu *et al.*[30] utilized global context information for head detection, Li *et al.*[18] used multilevel context information and segmentation results for object detection. Context is also important in dense classification problems such as semantic segmentation. For example, Chen *et al.*[5] encoded object size, ground plane and depth information into an energy function and minimized it to perform object detection. Similarly, Chen *et al.*[4] utilized object shape priors, the ground plane and semantic segmentation information to enhance detection accuracy, and Pham *et al.*[23] exploited depth features that included a disparity map and distance to the ground to propose objects. Lin, Zhao *et al.*[19,34] utilized context information by applying multiple pooling layers to guide pixel-wise classification. In particular, Hong *et al.*[12] proposed a detection method based on a co-occurrence context, and Lin *et al.*[19] used a chain of multiple max pooling blocks consisting of one max-pooling layer and one convolutional layer to extract features from large regions. Zhao *et al.*[34] used various pooling sizes to harvest both local and global context information and concatenate them to form the final feature representations.

In this paper, we propose a unified framework called SCAN that utilizes both multilevel middle features and context information to enhance object detection precision. More concretely, SCAN consists of two components, LFM and CFM. LFM fully explores the rich middle-feature maps, and constructs finer and more expressive feature maps to enhance object location, the most related work is Kong,Shrivastava and Lin *et al.*[15,28,20], while Kong *et al.*[15] only consider the fusion of multi-layer features, but did not use high-level features to improve the semantics of low-level features, Shrivastava *et al.*[28,20] using high-level features to improve the semantics of low-level features, but deal with them separately. In design LFM, we enhance the low-, middle-level features by high-level features and composite them to improve the detection accuracy. CFM is designed to utilize different scales of context information in the feature maps, the most related work is Lin *et al.*[19], which used a chain of multiple max pooling blocks and examined in the task of semantic segmentation. In designing CFM, we examined max pooling and average pooling in detail, and found that max pooling achieves more accurate detection, while average pooling achieves higher recall. Consequently, alternately employing max pooling and average pooling results in more balanced detection. To the best of our knowledge, this is the first effort to utilize both types of information to enhance object detection precision.

3. Method

3.1. Semantic Context Aware Network

Our method is based on two key observations. The first is that there is an internal conflict between the tasks of object detection and image classification. Object detection requires accurate object location information, while image classification must be insensitive to object translation, scaling and rotation. Consequently, the high-level features used in classification tasks are not suitable for accurately locating objects—especially small objects. Middle-level features contain more accurate location information but carry less semantic information. To remedy this problem, we use LFM to construct features that are both fine-grained and semantically rich. LFM uses internal features to obtain additional object location information. LFM primarily uses a top-down flow with lateral connections to bottom up features and concatenates them to construct semantic fine-grained features. The constructed features carry more semantic information than the equiv-

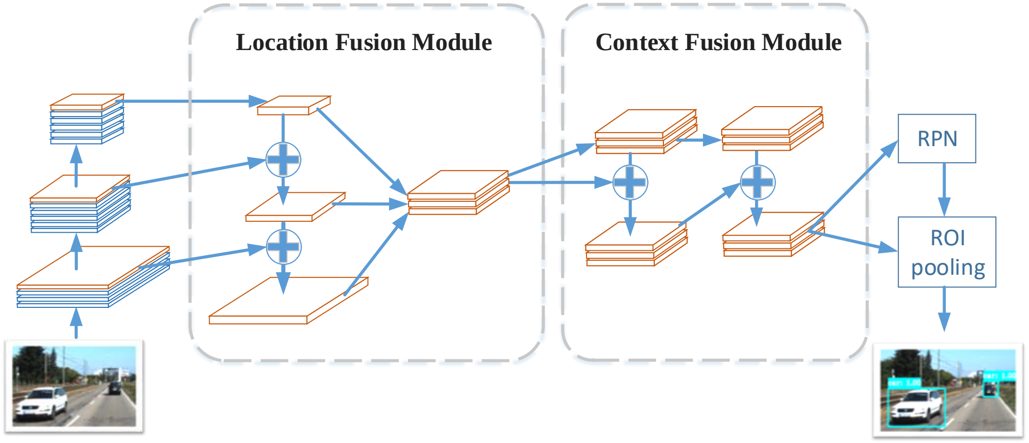

Fig. 1. Architecture of the proposed method. In the first step, a backbone convolutional network extracts multi-layer features from an image. Then, the Location Fusion Module constructs fine-grained and expressive feature maps using a top-down flow and lateral connections and the Context Fusion Module constructs context-aware features using multiple pooling operations. The final features are used to perform region proposal and classification.

alently sized bottom-up features while preserving more fine-grained object location information compared with higher-level features. Then, we utilize multilevel local context information to detect small objects and partially occluded objects. We designed CFM, which constructs context-aware features using a pyramid pooling operation. The context information improves the object detection accuracy, especially for small or partially occluded objects. Finally, the LFM and CFM models are combined into SCAN, forming a general-purpose framework suitable for multiple tasks. Fig. 1 shows the architecture of our method. The design details of the LFM and CFM modules and how they can be combined with a convolutional neural network for object detection are introduced in the following subsections.

3.2. *Location Fusion Module*

The goal of LFM is to construct features with higher resolution while preserving useful semantic information. LFM accomplishes this by merging bottom-up hierarchical feature maps using a top-down flow with literal connections, compressing them into a uniform space. In detail, given an image, we first apply the feed-forward computation of the backbone convolutional network to obtain bottom-up feature maps, most commonly at several scales and with numerous feature maps at the same scale. According to Lin *et al.*[20], we call a group of feature maps that maintain the same scale as a "stage," and we denote the output of these stages as $C_i, i = 1, , m$, where m is the length of the stages. We set $U_m = C_m$; then, we compute U_{i-1} by up-sampling U_i with a deconvolutional layer and merge the result with the feature map C_{i-1}:

$$U_{i-1} = C_{i-1} + Deconv(U_i). \tag{1}$$

This process is performed iteratively until the last layer is reached. In the experiments performed for this paper, our network used five stages; we used the final three stages of the feature maps, denoted as U_3, U_4, U_5. We applied a max pooling layer to the U_3 stage to construct the down-sampled feature map, and applied a deconvolutional layer to U_5 to construct an up-sampled feature map of the same size. Finally, we applied a 3x3 convolution layer and a batch normalization layer[14] to each feature map before concatenating them to obtain the final feature maps. Fig. 2 depicts the structure of the LFM.

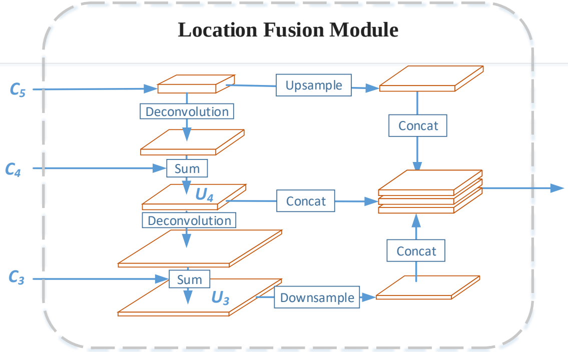

Fig. 2. The structure of the Location Fusion Module, which initially takes bottom-up features as input. Then, it takes a top-down flow started by deconvoluting the C_5 feature map with a scale of 2 and adds the corresponding bottom-up features of C_4 to compute U_4. Next, it computes U_3 by deconvoluting U_4 plus C_3. Finally, it concatenates the up-sampled C_5, down-sampled U_3, and U_4 to obtain the output features. For space reasons, the 3x3 convolution layer that occurs before the concatenation layer is not depicted.

3.3. *Context Fusion Module*

In a deep neural network, the size of the receptive field can roughly indicate the size of the context information; however Zhou *et al.*[35] shows that the empirical receptive field is smaller than the theoretical field. CFM is focused on adding different scales of context information into feature maps. It does this by fusing multiple pooling layers into the original feature maps. Mathematically, let C_i denote the CFM input, which is the output feature map from LFM. Here, C_{pj} denotes the j-th pooling from the feature map. Then,

$$C_o = \sum_j \phi(C_{pj}, C_{ij}). \tag{2}$$

where $\phi(\cdot)$ is the fusion method that can be a *sum* or a *concatenate* operator, and p can be either max pooling or average pooling. A trick exists to obtain a larger pooling context by repeatedly applying the same kernel size to the feature map—a technique developed by Zhao *et al.*[34] and adopted in our CFM implementation. From experiments, we found that using the *sum* operator for ϕ achieves more accurate detection results then does the *concatenate* operator; consequently, we adopted the *sum* operator. To choose a pooling layer, we found that max pooling results in more accurate detection results, but average pooling results in higher recall; therefore, we alternatively apply max pooling and average pooling to achieve more balanced detection results. Details of the process steps are illustrated in Fig. 3.

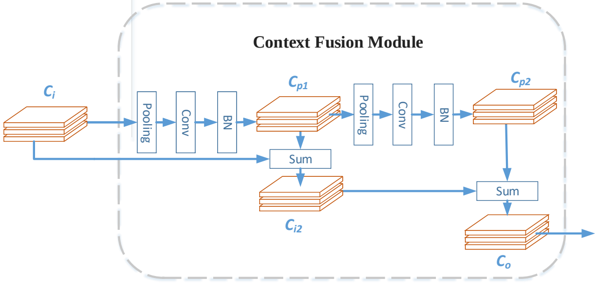

Fig. 3. The Context Fusion Module structure, which takes the output features from the LFM (denoted as C_i) and passes them through a pooling layer with a stride of 1 and a kernel size of 5. We obtain C_{i2} by summing the feature map C_{p1} and C_i. Then, we obtain C_{p2} by applying the pooling operation to the feature map C_{p1}. Finally, we sum C_{p2} and C_{i2} to obtain the final output feature map C_o.

4. Experiments

In this work, we focused on applying our method to the autonomous driving system; however, our approach is applicable to other object detection scenarios. We evaluated our model on the KITTI object detection dataset. The KITTI dataset consists of 7,481 training images and 7,518 test images containing a total of 80,256 labeled objects. The object labels are grouped into easy, moderate and hard levels, based on the extent to which the objects are occluded and truncated. The size of the images is 1,382 x 512. We randomly split the training images in half, forming a training set and a validation set. We evaluated our average precision results evaluated on the validation set.

We selected PVANET[13] as our baseline method. PVANET is a fast implementation of the Faster R-CNN detection protocol that uses a lighter back-end network structure. The network backbone is pre-trained on the ImageNet 1K classification set and fine-tuned on the KITTI training dataset. We use

Table 1. The results of ablation experiments from the PVANET (PVA) and our method with different combinations of modules on the KITTI validation set. Our model is trained based on the previous model to speed up network convergence (e.g., LFM training is based on the network weights of PVANET(PVA), and LFM+CFM is trained based on LFM. We applied the same hyperparameters and training iterations as were used previously (in %).

Model	Diff	Car		Cyclist		Pedestrian		mAP	mAR	FPS
		AP	AR	AP	AR	AP	AR			
PVA	E	95.56	98.84	88.7	95.38	85.6	88.61	81.23	84.45	**7.7**
	M	81.03	86.67	85.59	87.84	77.05	78.44			
	H	69.68	69.67	78.91	83.8	68.98	70.82			
LFM	E	95.8	97.61	88.4	95.08	86.3	88.92	81.68	84.36	7.3
	M	81.1	86.15	86.2	88.01	77.5	78.39			
	H	70.3	69.5	79.5	84.13	70	71.47			
CFM	E	86.42	96.93	88.29	95.38	79.43	87.03	76.52	81.87	6.2
	M	78.13	84.66	79	83.72	70.04	74.99			
	H	67.87	67.87	77.98	79.5	61.51	66.77			
SCAN	E	96.68	97.92	87.88	94.15	87.16	88.13	**81.96**	**84.71**	6.2
	M	80.65	85.74	86.65	89.8	78.22	78.6			
	H	70.42	69.89	79.7	86.45	70.28	71.73			

Stochastic Gradient Descent with momentum to optimize the loss function, with a weight decay of 0.0002 and a momentum of 0.9. The learning rate was set to 0.001 for the first 50K mini-batches and to 0.0001 for the next 50K iterations. To augment the training data, we resized every batch of image width to the value randomly selected from the sequence (480, 512, 554, 576, 608, 640, 672, 704, 736, 768) while maintaining the image ratio in the training phase. Then, in the testing phase, we rescaled the image widths to 768 pixels while maintaining the image ratios. We used a batch size of 1 and 512 anchors per image; our RPN uses 25 anchors at 5 scales (16, 32, 64, 128, and 256) and 5 aspect ratios (0.5, 0.667, 1.0, 1.5, and 2.0). We trained our model to detect 3 categories of objects: car, cyclist and pedestrian. We adopted a class-balanced sampling strategy so that every class would have a similar number of samples during the training process. All the experiments were performed using a Maxwell-based NVIDIA TITAN X GPU.

4.1. Ablation Experiments

We selected mean average precision (mAP), mean average recall (mAR), and frames per second (FPS) as the evaluation metrics in our experiment. mAP denotes the average precision scores for each object detection; it evaluates how good the detection result is. Following the KITTI estimate criteria, cars require more than a 70% overlap with the ground truth box, while pedestrians and cyclists require an overlap of 50% to be true positives. mAR denotes the average recall value from each category; it evaluates how many true positive objects were detected among all the detected objects. FPS specifies the number of images that can be detected in one second. For all three indicators, higher values indicate better performance.

As shown in Table 1 (PVA), the baseline PVANET model achieves a mAP of 81.23% on the validation set. Table 1 (LFM) shows that the detection accuracy was significantly improved after adding the LFM to PVANET, achieving a 0.46 improvement compared with the baseline. The results of this experiment show that LFM effectively improves both detection precision and recall, especially for detecting small objects such as the “hard” level of pedestrians and cyclists. Table 1 (CFM) shows the results from adding CFM directly to PVANET; the detection results are worse (the detection mAP drop down to 76.52) when CFM is added directly to PVANET. We argue that this result occurs because the feature map from the baseline method is not strong enough; consequently, it is difficult to obtain sufficient context information from CFM while

Table 2. Comparison experiments between our method(SCAN) and Faster R-CNN + OHEM(FRCNN): object detection results on the KITTI validation set (in %).

Model	Diff	Car		Cyclist		Pedestrian		mAP	mAR
		AP	AR	AP	AR	AP	AR		
FRCNN	E	93.5	97.92	49.16	57.85	64.22	68.12	60.13	64.87
	M	78.51	85.82	44.84	49.91	50.06	56.11		
	H	68.22	72.02	44.28	47.77	48.37	49.29		
SCAN	E	96.68	97.92	87.88	94.15	87.16	88.13	**81.96**	**84.71**
	M	80.65	85.74	86.65	89.8	78.22	78.6		
	H	70.42	69.89	79.7	86.45	70.28	71.73		

avoiding a jamming effect in the pooling operation.

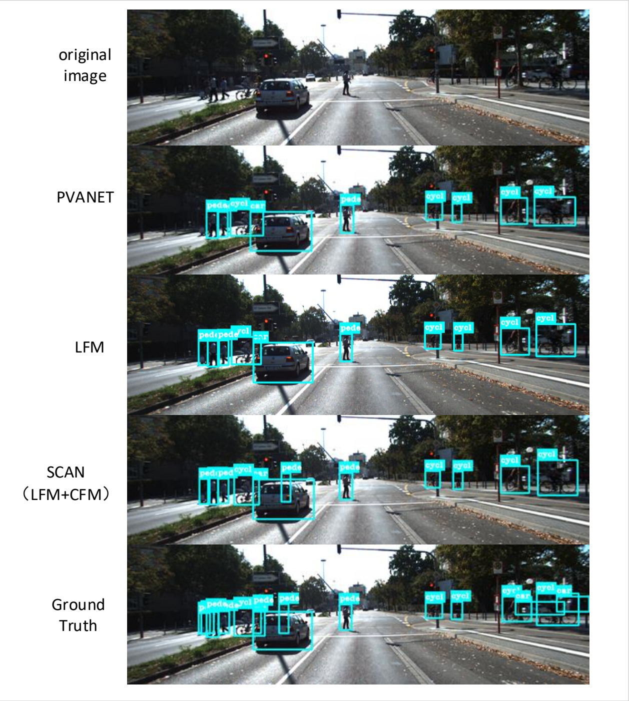

Fig. 4. An example of an original image from the KITTI dataset, the detection results and the ground truth. The first row shows the original KITTI image, the second row shows the baseline (PVANET) detection result, the third row shows the detection results of the baseline method with LFM, which achieves slightly more accurate bounding box locations than the baseline, and the fourth row shows the result of the PVANET+LFM+CFM method, which additionally detects the more difficult partially occluded objects. The final row shows the ground truth bounding boxes.

Table 1 (SCAN) shows the results from adding both LFM and CFM to PVANET, which further improves the mAP (from 81.68% to 81.96%) and achieves a mAR of 84.71%. This result indicates that the feature map output from LFM is sufficiently robust; therefore, CFM can successfully add the context information to the feature map and increase the detection precision and recall.

Fig. 4 shows the detection results of an example image from the validation set. We selected this image because it contains many types of objects to be detected and the object sizes and occlusions are rich. The first row shows the original image; the second row shows the results of PVANET (the baseline method); the third row shows the detection result of PVANET with LFM, demonstrating slightly more accurate bounding box locations compared with PVAN *et al.*; and the fourth row shows the results from the PVANET+LFM+CFM method, which detects more difficult objects such as those that are small or partially occluded. As Fig. 4 shows, the LFM+CFM method detects the pedestrian who is mostly occluded by the car. The final row shows the ground truth bounding boxes. Fig. 5 shows the precision and recall of PVANET, LFM and LFM+CFM. Again, LFM+CFM achieves the best results.

Faster R-CNN is one of the most successful object detection methods; therefore, we chose this method as the comparison method. We added OHEM to Fast R-CNN method because OHEM has been shown to be able to significantly improve Fast R-CNN's detection accuracy. We further improved the detection precision by removing all the layers from conv5 and using conv4_3 as the input for RPN as suggested by Zhang *et al.*[33], We trained this Faster R-CNN with OHEM[27] on the KITTI training set.

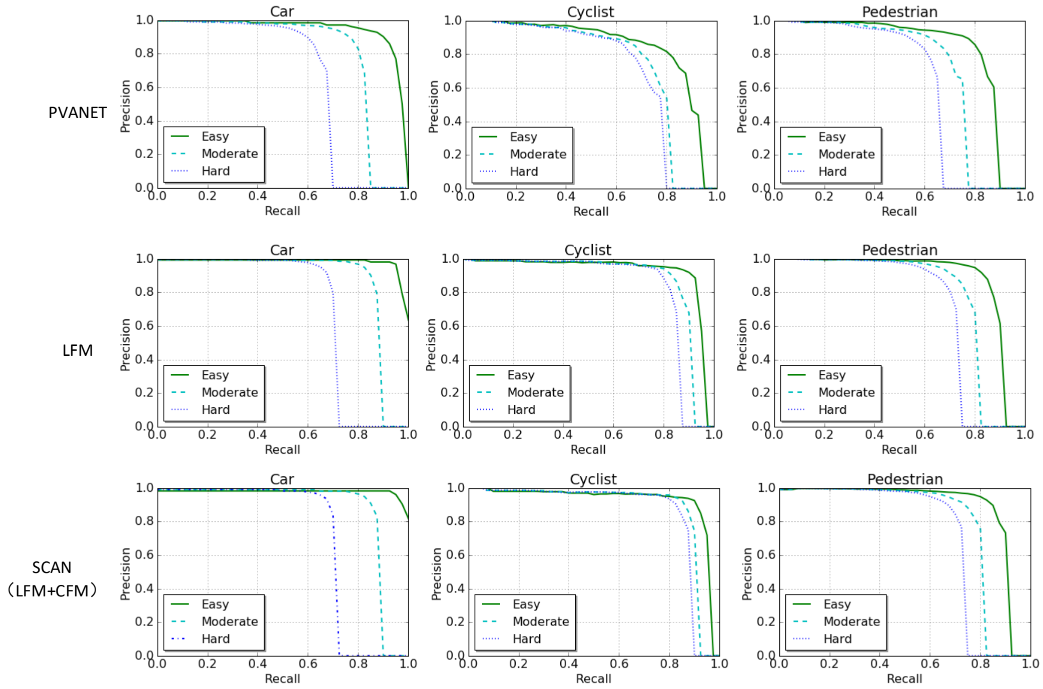

Fig. 5. Precision and recall of PVANET (the baseline method), combined with LFM and with LFM+CFM. The object detection results were obtained from the validation image set from the KITTI dataset.

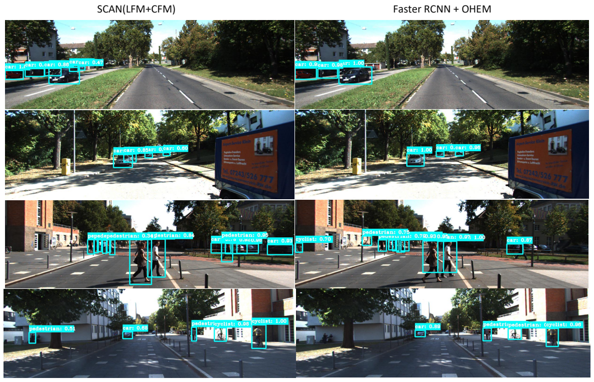

Fig. 6. Detection results from SCAN and Faster R-CNN + OHEM.

To make a fair comparison between Faster R-CNN with OHEM and our method, the RPN setting uses 25 anchors at 5 scales (16, 32, 64, 128, and 256) and 5 aspect ratios (0.5, 0.667, 1.0, 1.5, and 2.0), and we used the same training and validation sets. The detection results from our method with those from Faster R-CNN enhanced with OHEM are shown in Table 2. The results show that our method is significantly better than the Faster R-CNN + OHEM method, especially in detecting small objects such as pedestrians and cyclists. Fig. 6 shows some of the detection results from our method and Faster R-CNN+OHEM, revealing that our method detects more objects at the "hard" level and results in more accurate bounding box locations. Table 2 shows the object detection results of SCAN and Faster R-CNN + OHEM on the KITTI Dataset (in %).

4.2. *CFM Design Choices*

We designed experiments to choose between using max pooling and/or average pooling in CFM. Lin *et al.*[19] used a max pooling chain to fuse context information into feature maps for dense classification problems, while Zhao *et al.*[34] argued that using average pooling achieves better accuracy for semantic segmentation tasks.

We evaluated max pooling, average pooling and a mixture of both in CFM for object detection and found that while max pooling resulted in a slightly

Table 3. Ablation experiments on the selection of pooling type in designing the CFM and object detection results on the KITTI validation set (in %)

Pooling	Diff	Car		Cyclist		Pedestrian		mAP	mAR
		AP	AR	AP	AR	AP	AR		
Max	E	96.68	97.92	87.88	94.15	87.16	88.13	**81.96**	84.71
	M	80.65	85.74	86.65	89.8	78.22	78.6		
	H	70.42	69.89	79.7	86.45	70.28	71.73		
Average	E	95.96	97.71	87.23	96	87.1	90.43	81.68	**85.8**
	M	80.62	85.38	86.1	90.88	78.45	81.23		
	H	70.3	69.51	78.99	87.11	70.33	73.98		
Max + Average	E	95.47	97.65	88	95.38	86.98	89.64	81.78	85.12
	M	80.61	85.6	86.25	89.45	78.4	80.14		
	H	70.32	69.64	79.46	85.95	70.57	72.64		

higher object detection precision than the other options, average pooling resulted in a higher recall. The mixture of the two is a compromise choice; average pooling is a good candidate selection in some scenarios because it results in a mAP loss of only 0.28%; however, average pooling increased the mAR by 1.1% as shown in Table 3.

The second design choice decision for CFM was to determine how many pooling operations were appropriate. More pooling operations will create a large receptive field but can also result in a side effect of jamming. Therefore, we designed an experiment to test between one and four pooling layers. We found that two pooling layer achieved the best result. Three pooling layers obtains a similar detection result and achieves better results in detecting the moderate and hard levels of pedestrians. However, when we tested four pooling layers in CFM, both the mAP and the mAR decreased rapidly. This result indicated that the last pooling layer interfered with the feature map. The results are listed in Table 4.

Table 4. Ablation experiments on the selection of the number of pooling layers in the design of the Context Fusion Module: object detection results on the KITTI validation set (in %).

Pooling number	mAP	mAR
one pooling	81.28	84.66
two pooling	**81.96**	**84.71**
three pooling	81.79	84.67
four pooling	77.76	81.33

5. Conclusion

In this paper, we proposed SCAN, a novel network that combines precise location information and context information to enhance object detection accuracy for small and occluded objects in a simple way. There are two main modules in our network. The Location Fusion Module (LFM) obtains fine-grained semantic features and combines them to produce features that include precise positioning information. The Context Fusion Module (CFM) mixes contextual information into feature maps by applying multiple pooling layers to enhance the detection accuracy of small or partially occluded objects. Extensive experiments on the KITTI dataset show that our method enhances both object detection accuracy and recall, particularly for images containing small or partly occluded objects. We hope our work will be helpful in facilitating future research and applications.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. U1764261), the Natural Science Foundation of Shanghai (No.kz170020173571), the Fundamental Research Funds for the Central Universities (No. 22120170232) and Zhejiang Provincial Department of Education Project (No.Y201432357)

References

1. P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques and J. Malik. Multiscale combinatorial grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 328–335.
2. Z. Cai, Q. Fan, R. S. Feris and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision*, (Springer2016), pp. 354–370.
3. G. Chen, Y. Ding, J. Xiao and T. X. Han. Detection evolution with multi-order contextual co-occurrence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), pp. 1798–1805.
4. X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler and R. Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2147–2156.
5. X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler and R. Urtasun. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems* (2015), pp. 424–432.
6. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, (IEEE2005), pp. 886–893.
7. P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, **32** (9), (2010) 1627–1645.
8. R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1440–1448.
9. R. Girshick, J. Donahue, T. Darrell and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 580–587.
10. K. He, X. Zhang, S. Ren and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, (Springer2014), pp. 346–361.
11. K. He, X. Zhang, S. Ren and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 770–778.
12. J. Hong, Y. Hong, Y. Uh and H. Byun. Discovering overlooked objects: Context-based boosting of object detection in indoor scenes. *Pattern Recognition Letters*, **86**, (2017) 56–61.
13. S. Hong, B. Roh, K.-H. Kim, Y. Cheon and M. Park. Pvanet: Lightweight deep neural networks for real-time object detection. *arXiv preprint arXiv:1611.08588*, (2016).
14. S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, (2015).
15. T. Kong, A. Yao, Y. Chen and F. Sun. Hypernet: towards accurate region proposal generation and joint object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 845–853.
16. A. Krizhevsky, I. Sutskever and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105.
17. B. Li, T. Wu and S.-C. Zhu. Integrating context and occlusion for car detection by hierarchical and-or model. In *European Conference on Computer Vision*, (Springer2014), pp. 652–667.
18. H. Li, J. Brandt, Z. Lin, X. Shen and G. Hua. A multilevel contextual model for person recognition in photo albums. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 1297–1305.
19. G. Lin, A. Milan, C. Shen and I. Reid. Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation. *arXiv preprint arXiv:1611.06612*, (2016).
20. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie. Feature pyramid networks for object detection. In *CVPR* (2017).
21. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, (Springer2016), pp. 21–37.
22. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, **60** (2), 2004 91–110.
23. C. C. Pham and J. W. Jeon. Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks. *Signal Processing: Image Communication*, **53**, 2017 110–122.
24. S. Ren, K. He, R. Girshick and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (2015), pp. 91–99.
25. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, **115** (3), (2015) 211–252.
26. A. Shrivastava and A. Gupta. Contextual priming and feedback for faster r-cnn. In *European Conference on Computer Vision*, (Springer2016), pp. 330–348.

27. A. Shrivastava, A. Gupta and R. Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 761–769.
28. A. Shrivastava, R. Sukthankar, J. Malik and A. Gupta. Beyond skip connections: Top-down modulation for object detection. *arXiv preprint arXiv:1612.06851*, (2016).
29. J. R. Uijlings, K. E. Van De Sande, T. Gevers and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, **104** (2), (2013) 154–171.
30. T.-H. Vu, A. Osokin and I. Laptev. Context-aware cnns for person head detection. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 2893–2901.
31. Y. Xiang, W. Choi, Y. Lin and S. Savarese. Subcategory-aware convolutional neural networks for object proposals and detection. *arXiv preprint arXiv:1604.04693*, (2016).
32. F. Yang, W. Choi and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2129–2137.
33. L. Zhang, L. Lin, X. Liang and K. He. Is faster r-cnn doing well for pedestrian detection? In *European Conference on Computer Vision*, (Springer2016), pp. 443–457.
34. H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*, (2016).
35. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, (2014).
36. C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, (Springer2014), pp. 391–405.