IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 30, 2017, accepted February 16, 2018, date of publication March 12, 2018, date of current version April 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2815149

Object Detection Based on Multi-Layer Convolution Feature Fusion and Online Hard Example Mining

JUN CHU[1], ZHIXIAN GUO[2], AND LU LENG[1], (Member, IEEE)
[1]School of Software, Nanchang Hangkong University, Nanchang 330063, China
[2]School of Information Engineering, Nanchang Hangkong University, Nanchang 330063, China

Corresponding author: Lu Leng (leng@nchu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61663031, Grant 61741312, Grant 61772255, and Grant 61763033, in part by the Key Research and Development Project of Jiangxi Province under Grant 20171ACE50024 and Grant 20161BBE50085, in part by the Construction Project of Advantage Scientific and Technological Innovation Team under Grant 20165BCB19007, in part by the Application Innovation Program of Public Security Ministry under Grant 2017YYCXJXST048, in part by the Science and Technology Research Project of Education Department of Jiangxi Province under Grant GJJ150715, in part by the Open Foundation of Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition under Grant ET201680245 and Grant TX201604002, and in part by the Ph.D. Starting Foundation of Nanchang Hangkong University under Grant EA201620045.

ABSTRACT Object detection is a significant issue in visual surveillance. Faster region-based convolutional neural network (R-CNN) is a typical object detection algorithm of deep learning; however, neither its generalization ability nor its detection accuracy of small object is high. In this paper, an effective object detection algorithm is proposed for the small and occluded objects, which is based on multi-layer convolution feature fusion (MCFF) and online hard example mining (OHEM). First, the candidate regions are generated with region proposal network optimized by MCFF. Then, an effective OHEM algorithm is employed to train the region-based ConvNet detector. The hard examples are automatically selected to improve training efficiency. The avoidance of invalid examples accelerates the convergence speed of the model training. The experiments are performed on KITTI data set in intelligent traffic scenario. The proposed method outperforms the popular methods, such as Faster R-CNN, Regionlets, in terms of the overall detection accuracy. Furthermore, our method is good at the detection of small and occluded objects.

INDEX TERMS Deep leaning, multi-layer convolution feature fusion, object detection, online hard example mining, region proposal network.

I. INTRODUCTION

Object detection, as a remarkably important research field in computer vision, provides crucial information for the semantic understanding of image and video [1], [2]. Object detection is also employed in many other fields, like visual surveillance for public security [3], face detection and recognition [4]–[6], person re-identification [7], automatic drive [8], object detection in medical image [9], etc. Unfortunately, object detection suffers from several challenges, such as diversity of object scale, scene complexity, illumination variance, and occlusion [10], [11].

Object detection algorithms can be briefly categorized into two classes, namely classical methods and deep-learning-based methods [12]–[14]. Classical algorithms include sliding window selection [15], [16], manual feature design [17]–[19], and classifier design [20]. Firstly, the candidate regions are generated with sliding windows of different sizes, and then the features in the candidate regions are extracted by manual design. Finally, the classifiers are trained for detection. Deep-learning-based algorithms can be divided into region-free methods and region-based methods. The representative methods of the former include Single Shot MultiBox Detector (SSD) [21] and You Only Look Once (YOLO) [22], [23]; the representative methods of the latter include Region-based Convolutional Neural Network (R-CNN) [24], SPP-Net [25], Fast R-CNN [26] and Faster R-CNN [27].

Sliding windows were employed to extract candidate regions in classical object detection; however, the redundancy among the windows was substantial, which leaded to

This work is licensed under a Creative Commons Attribution 3.0 License. For more information, see http://creativecommons.org/licenses/by/3.0/

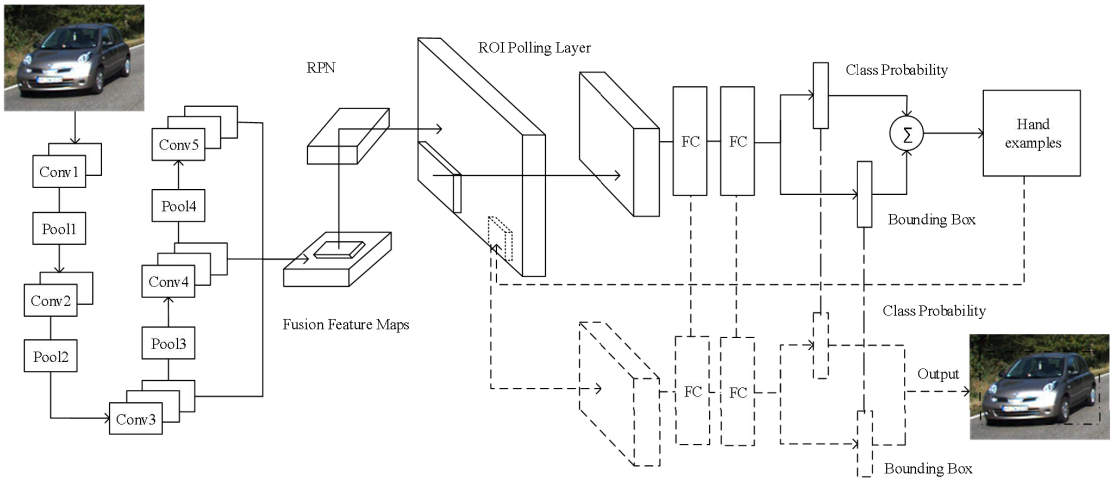


FIGURE 1. Framework of the proposed object detection algorithm.

high computational complexity. Selective search was used in R-CNN for object detection [28], instead of sliding window, to generate candidate windows more accurately. In addition, Convolutional Neural Network (CNN) outperforms manual design in robust feature extraction. The mean Average Precision (mAP) of R-CNN on Pascal VOC2010 data set [29] reached 53.7%; however, the speed was slow due to the convolution operations on all candidate regions. The input size of fully-connected layer had to be fixed, so the feature distortion and damage were unavoidable due to the cropping of candidate region.

Pyramid method was applied so that the input of any size was suitable for SPP-Net [25]. Moreover, the detection speed was accelerated because only one convolution was performed on the original image. The candidate boxes were mapped to the shared convolution layer of Fast R-CNN. The features in an image were extracted only once to speed up the detection; nevertheless, the selective search for candidate region generation in [26] was still time-consuming.

Region Proposal Network (RPN) was proposed for Faster R-CNN in [27] to directly generate candidate regions, which resulted in remarkable acceleration. An image of any arbitrary size could be input into RPN to generate a series of rectangular candidate regions with confidence levels. Fully-convolutional network in the RPN shared the convolution parameters with Fast R-CNN [30]. The convolution parameters in the first 5 layers were shared in Zeiler and Fergus' model (ZF) [31]; while those in the first 13 layers were shared in Simonyan and Zisserman's model (VGG-16) [32].

In order to generate candidate regions, a sliding window with the size of 3×3 sampled the shared convolution feature map of the last layer. Each sliding window was mapped to a low dimensional vector (256-dimension in ZF model, 512-dimension in VGG-16 model). These vectors were sent to the classification fully-connected layer and the regression fully-connected layer to obtain the category and location of the object. It is necessary to determine whether the objects are in the receptive field corresponding to each sliding window center. Multi-scale windows are required because both the sizes and height-width ratios of object are not uniform. RPN generated the reference window sizes, and then the sizes of "anchors", namely sliding windows, were adjusted according to the scales and height-width ratios. Faster R-CNN in [27] provided 9 reference windows and three scales so that 9 anchors were generated.

Faster R-CNN, as a state-of-the-art object detection algorithm, achieved the mAP of 70.4% on VOC 2012 data set. However, only the feature map generated on the convolution layer of Conv5_3 was input to RPN network, so it was not good at small object detection. Assume that the sizes of the input image and the feature map are 512×512 and 32×32, respectively. One point on the feature map is responsible for the feature expression of the surrounding area with the size of at least 16×16. Thus the features of small object cannot be expressed effectively.

In order to improve the accuracy of object detection, especially the detection of small objects, we propose a novel object detection algorithm based on multi-layer convolution feature fusion (MCFF) and online hard example mining (OHEM) [33]. The strengths of the proposed algorithm are prominent in terms of the detection of small and occluded objects with different scales. The experimental results on the Intelligent Transportation data set (KITTI) [34] confirm the advantages of our algorithm for the detection of "Car", "Pedestrian", and "Cyclist".

The rest of this paper is organized as follows: Section II elaborates the proposed algorithm. The experimental results and discussions are in Section III. Finally conclusions are drawn in Section IV.

II. METHODOLOGY

A. FRAMEWORK

Three principal types of objects, namely "Car", "Pedestrian", and "Cyclist", are ordinary in the scenario of

intelligent transportation, whose detection is frequently plagued with severe scale problem. High-level features contain low-resolution and high-semantic information. On the contrary, low-level features contain high-resolution and low-semantic information. Therefore, in this paper, high-level and low-level features are fused to address scale problem. Moreover, the feature maps, which are extracted with MCFF, are input to RPN network to generate candidate regions accurately. In the stage of object detection, OHEM is used to select the effective examples for the training of detection model. The framework of the proposed object detection algorithm is shown in Fig 1.

B. MULTI-LAYER CONVOLUTION FEATURE FUSION

The candidate region generated with RPN and the classification of candidate region are two main steps in Faster R-CNN. Obviously, the quality of candidate region is critical to object detection. The selective search in Fast R-CNN for candidate region generation results in computational complexity and time-consuming processing. RPN network, which is embedded in the entire CNN, shares the convolution features with the detection network so that the training speed is substantially accelerated.

Unfortunately, only the last layer of VGG-16, namely Conv5_3, is used as the input of RPN for feature extraction, which does not take into account both the pixel information and semantic information. The feature maps generated from low-level convolution contain more pixel information, which are helpful to the detection of small objects. In contrast, those generated from high-level convolution contain more semantic information, which are useful for the detection of large objects. Multi-layer data contain more complete information, which can detect the objects of different scales, so MCFF is effective for the extraction and detection of candidate regions.

In order to select good features for fusion, three models of VGG-16 network are trained on KITTI data set. KITTI contains 7,481 training images and 7,518 test images. Since none ground truth is provided in the test set, we divide the original training set into a training set and a test set with the quantity ratio of 7:3 according to [35] and [36]. The cross-validation method is used for the comparison. Conv3_3+Conv5_3, Conv4_3+Conv5_3, Conv3_3+Conv4_3+Conv5_3 are designed in Model 1, Model 2 and Model 3, respectively. "+" denotes fusing. All network parameters are configured identically.

Table 1 shows the detection accuracy by fusing different layers. The difficulty levels will be defined in Section III.D. Model 3, namely Conv3_3+Conv4_3+Conv5_3, outperforms the other models, and can better detect small objects. Fig. 2 shows the visualization of fusion feature, which not only reflects the response intensity of the feature map, but also shows the object location. The region of interest (ROI) commonly has a stronger response than the background.

Model 3, namely Conv3_3+Conv4_3+Conv5_3 fusion, is used to generate convolution fusion map. The framework of the fusion feature extraction is shown in Fig. 3. The scales

TABLE 1. Detection accuracy by fusing different layers (mAP, %).

	Easy	Med	Hard
		Cars	
5_3	80.35	69.58	58.27
3_3+5_3	82.36	70.15	63.56
4_3+5_3	82.54	71.01	62.89
3_3+4_3+5_3	**83.28**	**72.65**	**65.76**
		Pedestrians	
5_3	75.78	65.26	54.16
3_3+5_3	77.01	68.69	63.48
4_3+5_3	76.49	70.26	64.63
3_3+4_3+5_3	**77.28**	**71.43**	**65.20**
		Cyclists	
5_3	70.63	62.49	53.12
3_3+5_3	74.54	67.68	64.52
4_3+5_3	75.38	68.32	63.59
3_3+4_3+5_3	**76.31**	**69.98**	**65.81**

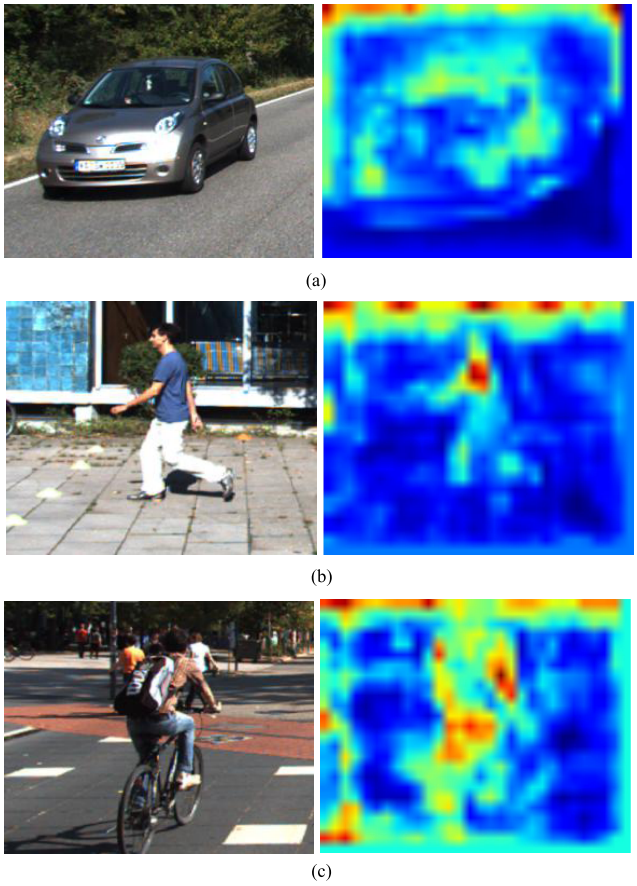


FIGURE 2. Fusion feature visualization. (a) Car; (b) Pedestrian; (c) Cyclist.

of the feature maps are not uniform, moreover, the size of the feature map is reduced with the layer level increment, so different sampling strategies are suitable for different layers. The sizes of the feature maps generated on Conv3_3 and Conv5_3 are converted into the size of Conv4_3. Maximum pooling sampling and deconvolution up-sampling are performed on the feature maps of Conv3_3 and Conv5_3, respectively. Local response normalization [37] is used to process the feature maps before fusion to improve the generalization capability. All new layers are initialized with "Xavier".

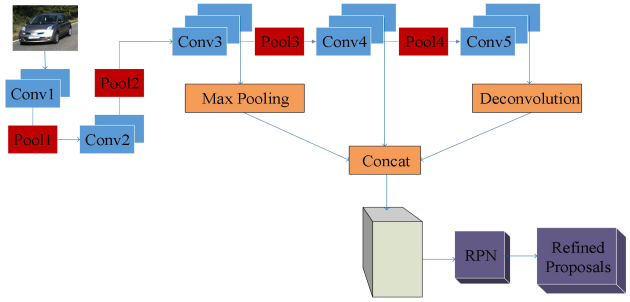


FIGURE 3. Fusion feature extraction.

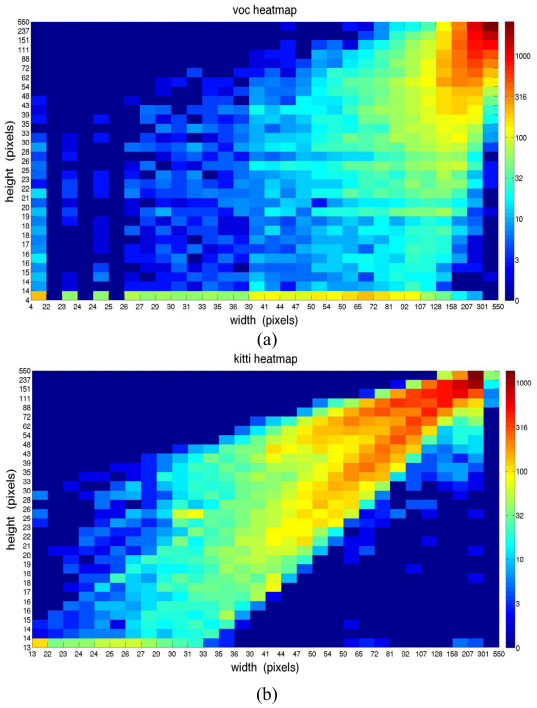


FIGURE 4. Size distribution of car object. (a) VOC2007; (b) KITTI.

9 anchors are used in RPN network, which are of 3 scales and 3 height-width ratios. Actually, they are the prediction boxes with 3 different areas (128^2, 256^2, 512^2) and 3 different height-width ratios (1:2, 1:1, 2:1). In order to accurately detect the objects of different scales, the anchor size should be as close as possible to the desired object size. Different from the objects in Pascal VOC data set, KITTI Vision Benchmark Suite data set contains more objects with different scales.

Fig. 4 compares the size distribution of car object in VOC and KITTI data sets. KITTI data set contains more small objects, even with the width of only 20∼60 pixels. Faster R-CNN predicts the anchor size based on VOC data set, whereas the prediction does not fit well with KITTI data set.

The extraneous information around real objects incurs detection deterioration. According to the coordinates of ground truth, the aspect ratios of three object types are shown in Table 2. In order to detect small objects and improve the overall detection, the anchor sizes are predicted with 6 different sizes (16^2, 32^2, 64^2, 128^2, 256^2, 512^2) and 5 different scales (1:2, 1:1, 2:1, 2.5:1, 3:1), which are suitable for the small objects.

TABLE 2. Aspect ratio of objects in KITTI data set.

Type	Aspect ratio				
Cars	1.8:1	2:1	2.2:1	2.5:1	3:1
Pedestrians	1:1.2	1:1.5	1:1.8	1:2	1:2.5
Cyclists	2:1	1.8:1	1:1	1:1.8	1:2

C. ONLINE HARD EXAMPLE MINING

Intersection-over-Union (IoU) between the anchor and ground truth is used to select the samples for RPN network training. The samples with the IoU greater than 0.7 and less than 0.3 are selected as the positive and negative examples, respectively. If none IoU is greater than 0.7, the sample with the maximum IoU is selected as the positive example. The candidate regions generated with region growing are mostly negative examples, so the quantity ratio between positive and negative examples is normally set to 1:3 to prevent the quantity imbalance between them. However, a large number of invalid negative examples are randomly selected, which causes the degradation of detection model.

The mini-batch in Faster R-CNN is set to 2 images, each of which generates 128 ROIs and feeds into ROI network. In fact, the RPN network in Faster R-CNN generates more than 128 ROIs, which are randomly selected from all ROIs. The quantity ratio between the positive and negative examples is 1:3, which indicates that the number of negative examples is larger than that of positive examples. The quantity imbalance and random selection of examples result in a poor expression capability and unsatisfactory detection of small object. Therefore, the hard examples with diversity and high loss are selected according to the loss values computed with OHEM. The selected examples are input to ROI network in way of back propagation. The architecture of OHEM is shown in Fig. 5.

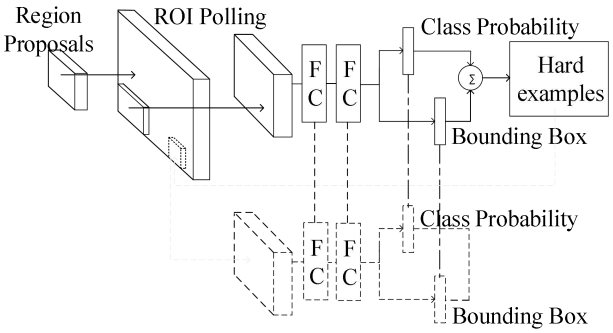


FIGURE 5. Architecture of OHEM.

The original Faster R-CNN is extended to two ROI networks that share the parameters. The parts connected with

FIGURE 6. Comparison of small object detection. (a) Original image; (b) Ground truth; (c) Faster R-CNN; (d) Fusion of RPN and MCFF; (e) Proposed algorithm; (f) Comparison between (e) and (b).

solid arrows constitute a read-only ROI network. All the operations in the read-only ROI network are forward, whose functions include ROI loss computation, ROI sequencing, and hard example selection. The output of the read-only ROI network predicts the classification result and the coordinates of the prediction boxes. Multi-task loss function is applied to optimize the minimum objective function. Multi-task loss function is defined as:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

The classification layer computes p_i, the probability of correct prediction of object class, where i denotes the index of the anchor in a mini-batch. When the anchor predicts a positive example, the probability of the ground-truth tag $p_i^* = 1$; for a negative example, $p_i^* = 0$. $t_i = \{t_x, t_y, t_w, t_h\}$ is a vector that represents the 4 parameterized coordinates of the predicted bounding box. t_i^* is the coordinate vector of the ground-truth corresponding to a positive anchor.

$L_{cls}(p_i, p_i^*)$ denotes the logarithmic loss of object prediction:

$$L_{cls}(p_i, p_i^*) = -\log[p_i^* p_i + (1 - p_i^*)(1 - p_i)] \quad (2)$$

$L_{reg}(t_i, t_i^*)$ denotes regression loss:

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) \quad (3)$$

where R is smooth $L1$ function:

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \quad (4)$$

The regression function is not considered if the classification result is background, i.e., $p_i^* = 0$. Classification and regression layers are normalized with N_{cls}, N_{reg} and equilibrium coefficient λ. When $\lambda = 10$, the normalized size of classification layer, N_{cls}, is the same as that of mini-batch, i.e., $N_{cls} = 256$. The normalized size of regression layer, N_{reg}, is the quantity of anchor, i.e., $N_{reg} \approx 2400$.

The examples are sorted and selected. The other ROI network contains both forward and backward operations. The inputs of this ROI network are hard examples. The loss values are computed and the gradients are propagated backward.

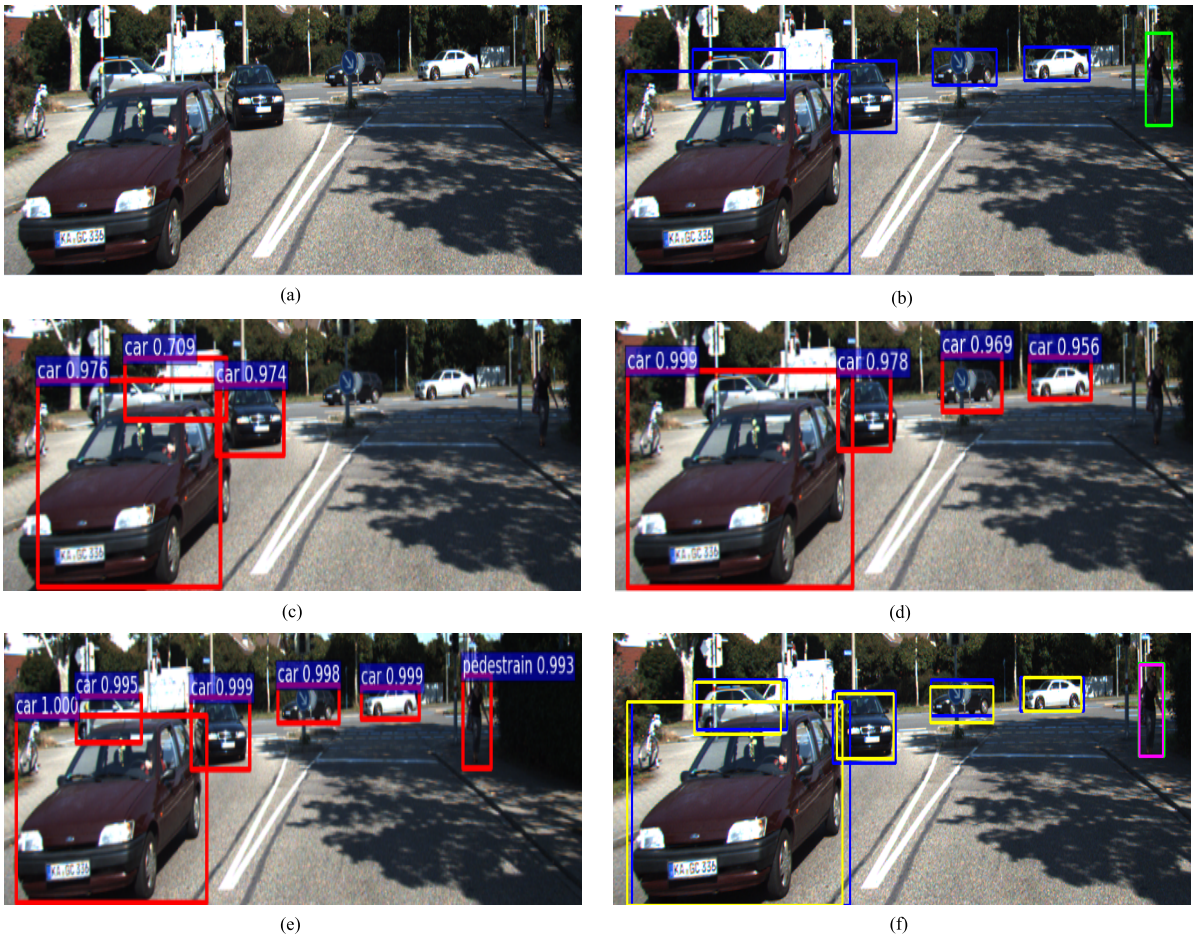


FIGURE 7. Comparison of occluded object detection. (a) Original image; (b) Ground truth; (c) Faster R-CNN; (d) Fusion of RPN and MCFF; (e) Proposed algorithm; (f) Comparison between (e) and (b).

This algorithm does not need to set the quantity proportion between the positive and negative examples to solve the quantity imbalance problem, so it eliminates the heuristic hyper-parameters. OHEM is more well-directed and further improves the accuracy of object detection.

III. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. EXPERIMENTAL SETUP

The experiments are performed on the PC with a CPU, i. e., Intel Xeon (R) E5-2603 including 6 cores (1.60GHz), 16GB RAM, and 64-bit Ubuntu 14.04. An NVIDIA GeForce GTX 1080 GPU is used for CNN computation.

B. DATA SET AND EVALUATION CRITERIA

The detection models are trained and the experiments are performed on the benchmark data set, namely KITTI data set in intelligent traffic scenario. KITTI, as an evaluation platform developed by Karlsruhe Institute of Technology, Germany, and Toyota Institute of Technology in Chicago, USA, is used to evaluate the performance of object detection and other computer vision technologies in practical complex environments.

Different from the object detection in common scenarios, like PASCAL VOC, etc., most objects in KITTI data set are medium-size and small objects. The minimum width of the object is only 25 pixels, and the maximum width exceeds 300 pixels, so the size range is wide. In addition, the occlusions between the objects are more complex. Moreover, the result is considered as correct only if the IOU between the detection box and the ground truth box is larger than 0.7. The aforementioned requirements undoubtedly increase the difficulty.

KITTI contains three object classes, namely Car, Pedestrian and Cyclist, and three evaluation levels, namely Easy, Moderate and Hard. We split the KITTI data set (7,481 images) into training set and test set with the quantity ratio of 7:3. It is found that the results do not change much if the quantity ratio approximates 7:3. At last, the number of training iteration is 10K.

C. QUALITATIVE RESULTS

The deep learning framework of Caffe in [38] is applied for training. The parameter weights of the pre-training model come from the results of VGG-16 trained on ImageNet, and

TABLE 3. Elaboration on difficulty level.

Difficulty	Object height	Occlusion	Allowable truncation area ratio
Easy	⩾40 pixels	None	[0, 15%]
Moderate	⩾25 pixels	Part	(15%, 30%]
Hard	⩾25 pixels	Large-area	(30%, 50%]

then are tuned slightly according to the specific detection tasks. The object detection model is re-trained in intelligent traffic scenario.

1) THE DETECTION OF SMALL OBJECT

Fig. 6 shows the experimental results of different algorithms for small object detection. (a) is the original image that contains several small objects. (b) is the ground truth. (c) is the result of the original Faster R-CNN algorithm, in which some small objects are not detected successfully. (d) is the result of RPN fused with MCFF, in which the detection performance is improved but a few small objects are still missed. (e) shows the result of the proposed algorithm. (f) compares the detected object boxes in (e) labeled in yellow and the ground truth boxes labeled in blue. All the small cars are correctly detected with our algorithm. Furthermore, the object localization is accurate. Thus our algorithm outperforms the compared methods in terms of small object detection.

2) THE DETECTION OF OCCLUDED OBJECT

Fig. 7 shows the experimental results of different algorithms for occluded object detection. (a) is the original image that contains 5 small cars and 1 pedestrian. The cars are occluded by the traffic signs or other cars. (b) is the ground truth box. (c) is the result of the original Faster R-CNN algorithm, in which some occluded objects are not detected successfully, and the pedestrian is not detected neither. (d) is the result of RPN algorithm fused with MCFF, in which the detection performance is improved but a few objects are still missed. (e) shows the result of the proposed algorithm. (f) compares the detected and the ground truth boxes. The ground truth boxes of ''car'' and ''pedestrian'' are labeled in blue and purple, respectively; while the detected object boxes of ''car'' and ''pedestrian'' are labeled in yellow and green, respectively. The distant cars, occluded cars, and the pedestrian are all correctly detected with our algorithm, and the object localization is accurate. Therefore, our algorithm outperforms the compared methods in terms of occluded object detection.

D. QUANTITATIVE RESULTS

Accuracy is computed to evaluate the proposed algorithm, and accordingly Precision-Recall curve is plotted. Three evaluation levels, namely ''easy'', ''moderate'' and ''hard'', are elaborated in Table 3.

Fig. 8 shows the good object detection performance of the proposed algorithm. The number of ''Car'' is larger than those of ''Cyclist'' and ''Pedestrian'' in KITTI data set, accordingly

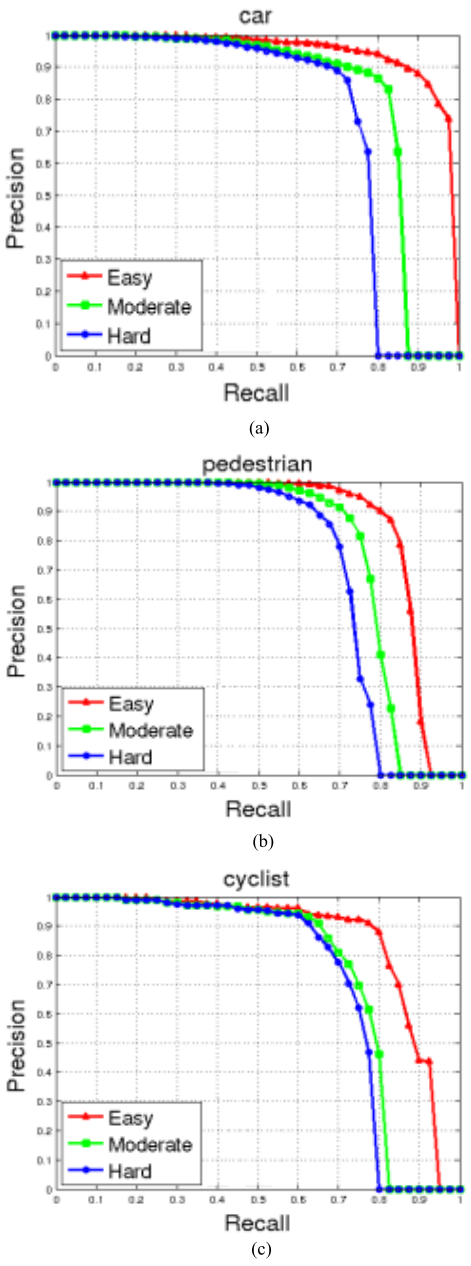


FIGURE 8. Precision-Recall curve of the proposed algorithm. (a) Car; (b) Pedestrian; (c) Cyclist.

the detection accuracy of ''Car'' is higher than those of ''Cyclist'' and ''Pedestrian'', which indicates that the data are important to the model training for object detection based on deep learning.

Table 4 compares the proposed algorithm with the popular object detection algorithms. Original Faster R-CNN, as a state-of-the-art algorithm, performs unsatisfactorily for ''Pedestrian'' and ''Cyclist'' with small sizes, but well for ''Car''. This indicates that the original Faster R-CNN is not very generalized, and does not work well for small objects. In this paper, more expressive features are extracted with MCFF. Furthermore, OHEM is used to mine more effective

TABLE 4. Performance comparison (PR, %).

Methods	Car			Pedestrian			Cyclist		
	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
SubCat [39]	84.14	75.46	59.71	54.67	42.34	37.95	--	--	--
pAUCEnsT[40]	--	--	--	65.26	54.49	48.60	51.62	38.03	33.38
Regionlets[41]	84.75	76.45	59.70	73.14	61.15	55.21	70.41	58.72	51.83
3DVP[42]	87.46	75.77	65.38	--	--	--	--	--	--
AOG[43]	84.80	75.94	60.70	--	--	--	--	--	--
Faster R-CNN[27]	87.90	**79.11**	70.19	78.35	65.91	61.19	71.41	62.81	55.44
DeepParts[]44	--	--	--	70.49	58.67	52.78	--	--	--
FilteredICF[45]	--	--	--	67.65	56.75	51.12	--	--	--
Ours	**88.50**	78.70	**70.36**	**82.25**	**75.36**	**69.90**	**83.07**	**73.73**	**69.10**

examples to train and generalize the model, so more small objects can be accurately detected, and accordingly the overall detection accuracy is improved. The algorithms, such as SubCat [39], pAUCEnsT [40] and Regionlets [41], are not suitable for small or occluded objects. The performances of Faster
R-CNN [27] and our algorithm are similar for "Car" detection, but our algorithm performs better for small and medium objects.

IV. CONCLUSIONS AND FUTURE WORKS

In this paper, a novel object detection algorithm is proposed based on multi-layer convolution feature fusion (MCFF) and online hard example mining (OHEM). The anchor sizes are adjusted according to the objects in the intelligent traffic scenario, so the small objects are detected accurately. Moreover, MCFF also improves the detection accuracy of small objects. OHEM is applies to mine more effective examples for training. The avoidance of invalid examples speeds up the convergence of the model training. The comprehensive comparison confirms the advantages of our algorithm. We will optimize classification network, detection speed and model size in our future works.

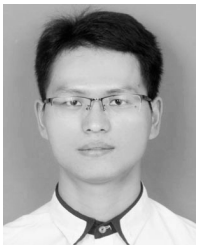
REFERENCES

[1] W. Ouyang *et al.*, "DeepID-Net: Deformable deep convolutional neural networks for object detection," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2403–2412.
[2] M.-C. Roh and J.-Y. Lee, "Refining faster-RCNN for accurate object detection," in *Proc. IEEE 15th IAPR Int. Conf. Mach. Vis. Appl.*, May 2017, pp. 514–517.
[3] W. Wang and M. Yao, "Intelligent transportation monitoring system based on computer vision," *J. Zhejiang Univ. Technol.*, vol. 38, no. 5, pp. 574–579, 2010.
[4] Z. Zheng and G. Guo, "A joint optimization scheme to combine different levels of features for face recognition with makeup changes," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2016, pp. 3001–3005.
[5] A. Rattani, D. R. Kisku, M. Bicego, and M. Tistarelli, "Feature level fusion of face and fingerprint biometrics," in *Proc. IEEE Int. Conf. Biometrics Theory, Appl., Syst.*, Sep. 2007, pp. 1–6.
[6] H. Proença, J. Neves, and J. Briceño, "Face recognition: Handling data misalignments implicitly by fusion of sparse representations," *IET Comput. Vis.*, vol. 9, no. 2, pp. 216–225, 2014.
[7] N. Aparajita, P. K. Sa, S. K. Choudhury, S. Bakshi, and B. Majhi, "A neuromorphic person re-identification framework for video surveillance," *IEEE Access*, vol. 5, pp. 6471–6482, 2017.
[8] J. Ajin, V. Jayanthi, and D. Baskar, "Automatic object detection in car-driving sequence using neural network and optical flow analysis," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res.*, Dec. 2015, pp. 1–4.
[9] L. Yao, J. Tian, and H. He, "Image segmentation via fuzzy object extraction and edge detection and its medical application," *J. X-Ray Sci. Technol.*, vol. 10, nos. 1–2, pp. 95–106, 2001.
[10] S. K. Choudhury, P. K. Sa, S. Bakshi, and B. Majhi, "An evaluation of background subtraction for object detection vis-a-vis mitigating challenging scenarios," *IEEE Access*, vol. 4, pp. 6133–6150, 2017.
[11] J. Chu *et al.*, "Target tracking based on occlusion detection and spatio-temporal context information," *Pattern Recognit. Artif. Intell.*, vol. 30, no. 8, pp. 718–727, 2017.
[12] H. Peng, B. Chen, Y. Cai, and Z. D. Liu, "Vision-based object detection and tracking: A review," *Acta Automatica Sinica*, vol. 42, no. 10, pp. 1466–1489, 2016.
[13] H. Zhang, K. Wang, and F. Wang, "Advances and perspectives on applications of deep learning in visual object detection," *Acta Automatica Sinica*, vol. 43, no. 8, pp. 1289–1305, 2017.
[14] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst. Curran Assoc. Inc*, 2012, pp. 1097–1105.
[15] F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[16] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 52, no. 2, pp. 137–154, 2004.
[17] B. Triggs *et al.*, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Comput. Soc.*, Jun. 2005, pp. 886–893.
[18] G. Cinbis, J. Verbeek, and C. Schmid, "Segmentation driven object detection with Fisher vectors," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2014, pp. 2968–2975.
[19] J. Wan, G. Guo, and S. Li, "Explore efficient local features from RGB-D data for one-shot learning gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1626–1639, Aug. 2016.
[20] G. Guo, S. Li, and K. Chan, "Support vector machines for face recognition," *Image Vis. Comput.*, vol. 19, nos. 9–10, pp. 631–638, 2010.
[21] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. Springer, Cham*, 2016, pp. 21–37.
[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
[23] J. Redmon and A. Farhadi. (Dec. 2016). "YOLO9000: Better, faster, stronger." [Online]. Available: https://arxiv.org/abs/1612.08242
[24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[25] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[26] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.
[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[28] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

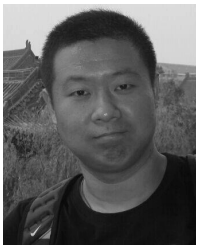
[29] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
[30] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.
[31] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.
[32] K. Simonyan and Z. Andrew. (Sep. 2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[33] S. Abhinav, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 761–769.
[34] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
[35] X. Chen *et al.*, "3D object proposals for accurate object class detection," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 424–432.
[36] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 354–370.
[37] A. E. Robinson, P. S. Hammon, and V. R. de Sa, "Explaining brightness illusions using spatial filtering and local response normalization," *Vis. Res.*, vol. 47, no. 12, pp. 1631–1644, 2007.
[38] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[39] E. Ohn-Bar and M. M. Trivedi, "Learning to detect vehicles by clustering appearance patterns," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2511–2521, Oct. 2015.
[40] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Pedestrian detection with spatially pooled features and structured ensemble learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1243–1257, Jun. 2016.
[41] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2017, pp. 17–24.
[42] X. Yu, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3D Voxel patterns for object category recognition," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1903–1911.
[43] B. Li, T. Wu, and S.-C. Zhu, "Integrating context and occlusion for car detection by hierarchical and-or model," in *Proc. Comput. Vis. ECCV*, 2014, pp. 652–667.
[44] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1904–1912.
[45] S. Zhang, R. Benenson, and B. Schiele, "Filtered channel features for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1751–1760.

JUN CHU received the Ph.D. degree from Northwestern Polytechnic University, Xi'an, China, in 2005. She was a Post-Doctoral Researcher with the Exploration Center of Lunar and Deep Space, National Astronomical Observatory of Chinese Academy of Sciences, from 2005 to 2008. She was a Visiting Scholar with the University of California at Merced, Merced, CA, USA.

She is currently the Director of the Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition, the Vice President, and a Full Professor with the School of Software, Nanchang Hangkong University. She was also a member of the Computer Vision Special Committee, China Computer Federation, and the Director of the Jiangxi Institute of Computer Science. Her research interests include computer vision and pattern recognition.



ZHIXIAN GUO received the bachelor's degree from Gannan Normal University, Ganzhou, China. He is currently pursuing the master's degree with Nanchang Hangkong University. His research interests include computer vision and image processing.



LU LENG (M'12) received the Ph.D. degree from Southwest Jiaotong University, Chengdu, China, in 2012. He did his post-doctoral research at Yonsei University, Seoul, South Korea, and the Nanjing University of Aeronautics and Astronautics, Nanjing, China. He was a Visiting Scholar with West Virginia University, USA. He is currently an Associate Professor with Nanchang Hangkong University.

He has authored or co-authored over 60 international journal and conference papers, and been granted several scholarships and funding projects in his academic research. He is the reviewer of several international journals and conferences. His research interests include image processing, biometric template protection, and biometric recognition.

Dr. Leng is a member of the Association for Computing Machinery and China Computer Federation.