IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 16, 2019, accepted June 6, 2019, date of publication June 12, 2019, date of current version June 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2922479

Multi-Scale Detector for Accurate Vehicle Detection in Traffic Surveillance Data

KWANG-JU KIM[1], PYONG-KUN KIM[1], YUN-SU CHUNG[1], AND DOO-HYUN CHOI[2]
[1]Electronics and Telecommunications Research Institute, Daegu 42994, South Korea
[2]School of Electronics Engineering, Kyungpook National University, Daegu 41566, South Korea

Corresponding author: Doo-Hyun Choi (dhc@ee.knu.ac.kr)

This work was supported by the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean Government (Development of ICT Convergence Technology for Daegu-GyeongBuk Regional Industry) under Grant 19ZD1100.

ABSTRACT The recent research by deep learning has shown many breakthroughs with high performance that were not achieved with traditional machine learning algorithms. Particularly in the field of object detection, commercial products with high accuracy in the real environment are applied through the deep learning methods. However, the object detection method using the convolutional neural network (CNN) has a disadvantage that a large number of feature maps should be generated in order to be robust against scale change and occlusion of the object. Also, simply raising the number of feature maps does not improve performance. In this paper, we propose to integrate additional prediction layers into conventional Yolo-v3 using spatial pyramid pooling to complement the detection accuracy of the vehicle for large scale changes or being occluded by other objects. Our proposed detector achieves 85.29% mAP, which outperformed than those of the DPM, ACF, R-CNN, CompACT, NANO, EB, GP-FRCNN, SA-FRCNN, Faster-R CNN2, HAVD, and SSD-VDIG on the UA-DETRAC benchmark data-set consisting of challenging real-world-traffic videos.

INDEX TERMS UA-DETRAC benchmark, traffic surveillance, deep learning, machine learning, neural networks, object detection, scale variation, occlusion, yolo.

I. INTRODUCTION

The demand of intelligent traffic surveillance system has been increased in order to improve traffic efficiency by preventing various traffic problems. For intelligent traffic surveillance, real-time traffic information should be obtain consistent and accurate information about trajectory of vehicles. Therefore, computer vision-based effective object detection methods that extract traffic information automatically from real-time video camera are very important for the reliable intelligent traffic surveillance system [1], [2]. Figure 1 shows the scene complexity of images from traffic security cameras to detect every vehicles due to large variations on object such as scales, types, perspectives, occlusion, lighting/brightness conditions and different weather conditions. In recent years, various deep learning based object detection models have applied to increase the reliability of intelligent traffic information acquisition. Among deep neural network based detection methods such as Faster R-CNN, SSD and Yolo-v3 [3]–[5], Yolo-v3 has a relatively fast and high mAP performance that is robust to scale variation and occlusion since it employs multiple convolution and prediction layers for multi-scale object detection. Although the Yolo-v3 performs good enough in both the speed and accuracy, there is still potential for improvement. Figure 2 shows overall architecture of the

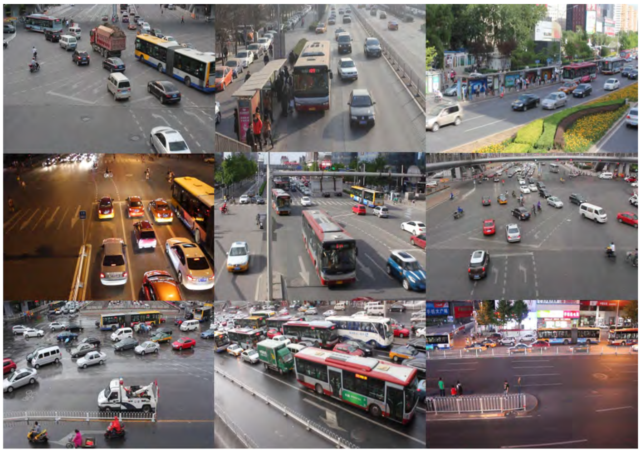


FIGURE 1. Sample images of the UA-DETRAC benchmark data-set [9] that contain various challenges on vehicle detection such as weather conditions, varying times, large scale changes and occlusions.

The associate editor coordinating the review of this manuscript and approving it for publication was Chunbo Xiu.

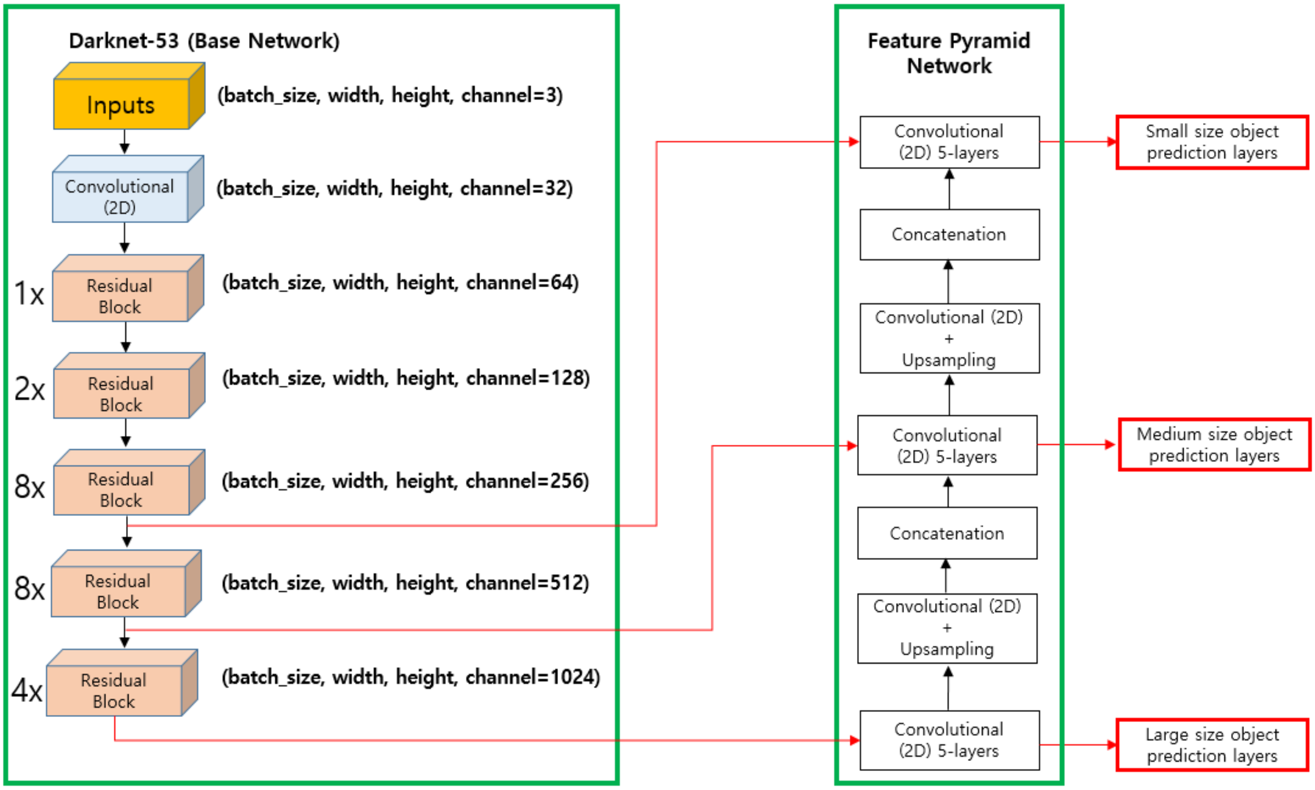

Darknet-53 (Base Network)
Inputs
(batch_size, width, height, channel=3)
Convolutional (2D)
(batch_size, width, height, channel=32)
1x
Residual Block
(batch_size, width, height, channel=64)
2x
Residual Block
(batch_size, width, height, channel=128)
8x
Residual Block
(batch_size, width, height, channel=256)
8x
Residual Block
(batch_size, width, height, channel=512)
4x
Residual Block
(batch_size, width, height, channel=1024)
Feature Pyramid Network
Convolutional (2D) 5-layers
Concatenation
Convolutional (2D) + Upsampling
Convolutional (2D) 5-layers
Concatenation
Convolutional (2D) + Upsampling
Convolutional (2D) 5-layers
Small size object prediction layers
Medium size object prediction layers
Large size object prediction layers

FIGURE 2. **Overall architecture of the conventional Yolo-v3 framework.**

Yolo-v3 framework. They provide a Darknet-53 network as base network that extracts feature from 3 different scales using a similar concept to feature pyramid network [6]. After through the feature pyramid network, it adds several 2D-convolutional layers as prediction layers. The last of 3 layers output the result a 3-d tensor encoding bounding box, objectiveness, and class predictions. However, Yolo-v3 has still a difficulty in detecting various sizes vehicles such as car, van, bus, and truck in real world image from traffic surveillance camera because the last layers support only three different scales of the objects. This paper is an extended version of work published in [7] that ranked in the 3rd place of 15th IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS 2018) & the International Workshop on Traffic and Street Surveillance for Safety and Security (IWT4S) Challenge on Advanced Traffic Monitoring. In this paper, we present a scale invariant vehicle detection network with spatial pyramid pooling method for more robust vehicle detection as follows. First, two more object prediction layers were inserted in the conventional Yolo-v3 framework. More specifically, one additional prediction layer is between large size object prediction layer and medium. Another additional prediction layer is between medium size object prediction layer and small one. Second, the spatial pyramid pooling (SPP) networks [8] were added before each prediction layer after through feature pyramid network. Our proposed method clearly outperforms the previous object detection methods on the UA-DETRAC benchmark data-set especially in case of crowd conditions. The remainder of this paper is organized as follows. Section II describes a brief overview about the background and related works in the area of deep learning based detection methods and traffic surveillance data. The proposed multi-scale vehicle detector with spatial pyramid pooling, a modified version of Yolo-v3, is presented in Section III. Section IV performs the experiments and results. Section V then presents our conclusions.

II. RELATED WORKS

Most recent deep learning based detection methods have been studied in order to overcome the problem of object detection and the research still keep continuing to improve performance. Before deep learning based object detection framework introduced, one of state of the art object detector was led by approaches exploiting Deformable Part-based Models (DPMs) [10]. DPMs successfully detects the target objects by finding the object parts and combining their spatial information. Aggregate Channel Features (ACF) detector [11] also achieved state of the are result of detecting scalable objects successfully by using of computing input image's multiple channels and sum every block of pixels from a multi-scale sliding window. After Alex-net won the challenge for visual object recognition called the ImageNet Large Scale Visual Recognition Challenge(ILSVRC) in 2012 [12], Convolutional Neural Network (CNN) showed a significant breakthrough in the field of object recognition and classification. Also, CNN have shown their powerful feature representative ability in the field of object detection. Some CNN based object detection approaches have been presented to aim at learning invariant CNN representations with respect to different types of transformations such as scale, rotation and both [13]–[15]. In general, these object detection based on deep learning frameworks can be divided into two categories, region proposal based frameworks (two-stage detector) and regression/classification (one-stage detector) based frameworks. The first consists of two stages. First, It candidates regions so called as a region proposal method and then classifies those regions in a subsequent classification. The Region proposal based frameworks like R-CNN [16], Fast R-CNN [17] and Faster R-CNN typically comprise above two stages. R-CNN began with object region proposals in an image by selective search [18] or Edge boxes [19] method and then performed the classification but led to large time latency. The Fast R-CNN was proposed to reduce the time consumption related to large number of region proposals. The feature maps size was reduced using a Region of Interests(RoI) pooling layer to acquire valid RoIs. These Region proposals method was still computationally expensive because of the selective search. Faster R-CNN introduced Region Proposal Network (RPN) to directly generate region proposals and predicted object locations. Based on Faster R-CNN framework, significant improvements have been deployed. For the one-stage object detector, OverFeat [20] was the earlier work. Among object detection frameworks such as SSD, Yolo-v1 [21], Yolo-v2 [22], and yolo-v3 that perform localization and classification at once are typical one-stage detection methods. SSD employs VGG-16 [23] as a base network that are applied on multiple feature maps to account for various object scales. The only different method of SSD and Yolo-v2 is that only one feature map is used for prediction and anchor box regression. Instead of VGG-16, the yolo-v2 authors proposed Darknet-19 as base network. The Darknet-19 uses mostly convolutional layers without the large fully connected layers at the end. This model performs in decreased inference time compared to VGG-16. To determine the anchor box dimensions, k-means clustering is employed. Yolo-v3 which is enhanced version of Yolo-v2, included multi-scale predictions and a better base network called as Darknet-53. The Darknet-53 has 53 convolutional lyaers which is significant larger than Darknet-19 and it consists of successive 3 × 3, 1 × 1 convolutional layers and several shortcut connections [24].

III. METHOD

In the following section, we describe our strategy which is to add two more object prediction layers and insert SPP-networks before each prediction layer to improve accuracy for the detection of strong scale variations and occlusions. At first, we describe the whole pipeline of the detection framework and additional prediction layers in section III.A. Then, we present the insertion of the SPP-network in section III.B. Finally, we present the replacement of non maximum suppression (NMS) to Soft-NMS [25] in the bounding box merge stage of Yolo-v3 in section III.C.

A. ADDITIONAL PREDICTION LAYERS

We employ a Darknet-53 network as a base network for feature extraction. The Darknet-53 has deeper convolution layers(53 convolution layers) than Yolo-v2 (19 convolution layers) and it also has residual blocks, shortcut connections, and up-sampling. From the Darknet-53 network, the feature maps are generated and then sent to the Feature Pyramid Network (FPN). Our proposed multi-scale vehicle detection architecture is based on the conventional Yolo-v3 detection framework. Figure 2, 3 shows the conventional yolo-v3 and our proposed architecture, respectively. From Figure 3, The multiple different things of the conventional Yolo-v3 and our proposed architecture are that 2 more prediction layers and 5 more SPP-networks with batch normalization [26] are used to account for various object scales. By adding 2 more prediction layers, our proposed architecture have the robustness to vehicle scale with a wide scale range of anchor. After customizing the scale of anchor box size using K-means clustering method, we design the 15 anchor boxes from 5 to 400 pixels based on the effective receptive field. The anchor boxes on early stage feature maps cover a smaller receptive field to detect objects at a smaller scale, but the anchor boxes on later stage feature maps cover a larger receptive field to detect objects with larger scale. The early stage convolution layers of a deep neural network have weak object information that only high-level features of an input image. To compensate lack of the object information, we combine features from different layers of early stage convolution and later stage convolution network. However, since feature maps from layers at different stages have different dimensions, we apply a up-sampling operation to combine them effectively. Then the combined feature maps is again subjected a few 1 × 1 convolutional layers to fuse the features from the earlier stage layer. A batch normalization layers is followed to receive the final feature map. The final feature map, which up-sampled layers concatenated with the previous layers, helps preserve the fine grained features helping in object detection. We apply this procedures on feature pyramid network before each prediction layers.

B. SPATIAL PYRAMID POOLING NETWORKS

Pooling layer is used to progressively reduce the dimension of feature representation from convolution layer, inserted in-between successive convolution layers. Typically three types of pooling layers are commonly observed (general pooling, overlapping pooling and SPP). The typical ways of general pooling includes max pooling and average pooling. The overlapping pooling usually set larger filter width than the stride. SPP can produce a single high-level feature vector of

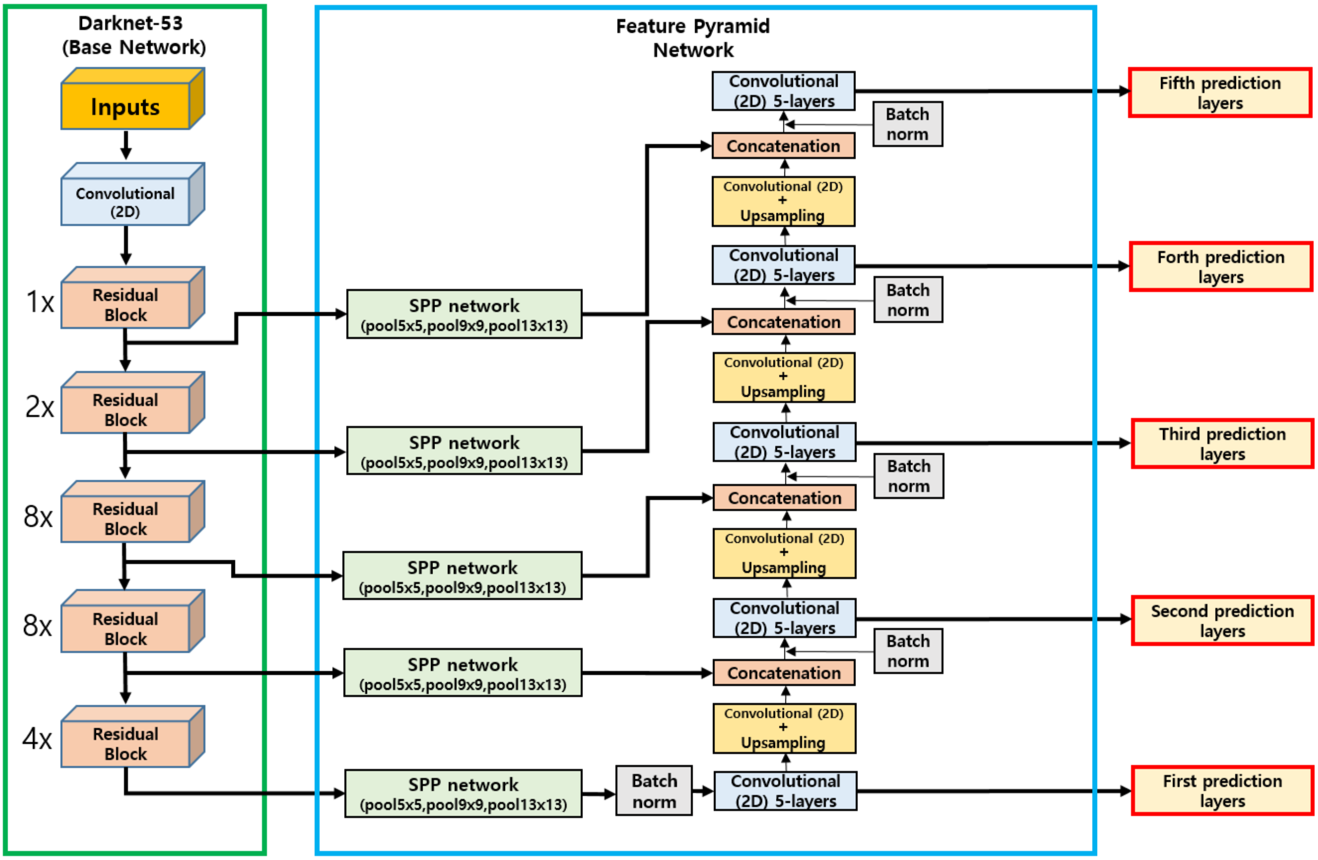

Darknet-53
(Base Network)
Inputs
Convolutional (2D)
1x
Residual Block
2x
Residual Block
8x
Residual Block
8x
Residual Block
4x
Residual Block
Feature Pyramid Network
SPP network (pool5x5,pool9x9,pool13x13)
SPP network (pool5x5,pool9x9,pool13x13)
SPP network (pool5x5,pool9x9,pool13x13)
SPP network (pool5x5,pool9x9,pool13x13)
SPP network (pool5x5,pool9x9,pool13x13)
Batch norm
Convolutional (2D) 5-layers
Batch norm
Concatenation
Convolutional (2D) + Upsampling
Convolutional (2D) 5-layers
Batch norm
Concatenation
Convolutional (2D) + Upsampling
Convolutional (2D) 5-layers
Batch norm
Concatenation
Convolutional (2D) + Upsampling
Convolutional (2D) 5-layers
Batch norm
Concatenation
Convolutional (2D) + Upsampling
Convolutional (2D) 5-layers
Batch norm
Fifth prediction layers
Forth prediction layers
Third prediction layers
Second prediction layers
First prediction layers

FIGURE 3. Proposed multi-scale detector for accurate vehicle detector based on Yolo-v3 model.

TABLE 1. Feature map concatenation from our proposed network.

Prediction	Feature Map Concatenation	
Layers	Base Network	Feature Pyramid Network
First	None	None
Second	61th with SPP-network	85th with up sampling(x2)
Third	36th with SPP-network	97th with up sampling(x2)
Forth	11th with SPP-network	109th with up sampling(x2)
Fifth	4th with SPP-network	121th with up sampling(x2)

images with any size, into the fixed size of dimensions. This advantage of SPP enables that the input image does not need to be cropped. Hence it can avoid the information loss caused by cropping and warping. In object detection field, the SPP-network preforms an input feature map with filters or pooling operations at different rates. These executions can generate multiple effective field-of-views. Table 1 shows where we apply SPP-network and how we concatenate feature maps in our proposed architecture. Since the feature values from different layers are quite different, a normalization step is needed before feature map concatenation. We applied batch normalization at each filters before concatenation.

C. SOFT-NMS

Non-maximum suppression (NMS) is widely used as a post-processing step in object detection frameworks to merge the nearby detection bounding boxes around one object. However, when two objects are highly overlaped each other, the detection bounding box with the lower score will be excluded, which harms the performance of object detection. Soft non-maximum suppression (Soft-NMS) treats the detection scores of all other objects as a function of IOU that possesses maximum score. So the detection bounding box with lower score would not be deleted directly. Hence, we apply the soft-NMS to replace the conventional NMS used in object detection to discount the confidence score of predicted boxes rather than completely discarding them. Also, Soft-NMS shows the better AP scores than conventional NMS for several benchmark.

IV. EXPERIMENTS

In this section, we evaluate the detection performance of our proposed detector. First, we introduce the UA-DETRAC benchmark data-set that is used for our experiments and describe the training details. Finally, we compare our detection accuracy and speed to state-of-the art detector results in traffic surveillance field. To evaluate the performance of our proposed approaches in the experiments, we used the mean average precision (mAP) score by taking precision-recall curve over intersection over union (IOU) at 0.7 threshold as given by the UA-DETRAC benchmark data-set

TABLE 2. Evaluation results of the proposed architecture trained by a different number of SPP-network.

Number of SPP-network	Overall	Easy	Medium	Hard	Cloudy	Night	Rainy	Sunny	FPS
0	82.37	94.63	87.02	72.68	86.93	82.27	75.57	89.03	20-22
1	83.30	94.60	88.81	73.07	86.99	85.96	74.97	90.73	17-19
3	84.29	95.73	88.37	75.45	87.65	85.80	78.03	90.14	13-14
5	85.29	96.04	89.42	76.55	88.00	88.67	78.90	88.91	9-10

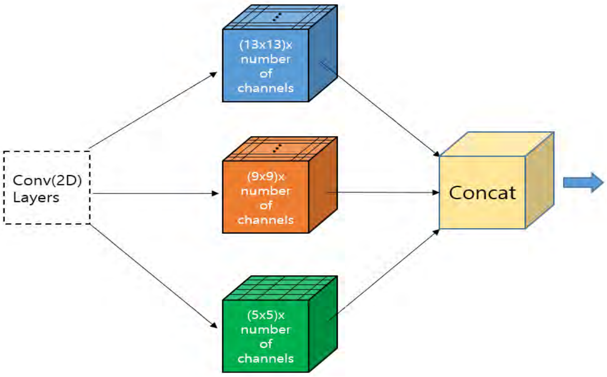

(13x13)x number of channels
Conv(2D) Layers
(9x9)x number of channels
Concat
(5x5)x number of channels

FIGURE 4. Implementation of spatial pyramid pooling on the proposed architecture. Before we concatenate two feature maps, we applied SPP-network in our base network.

Evaluation Protocol. Test data-set includes 10 easy sequences, 20 medium sequences, and 10 hard sequences.

A. UA-DETRAC BENCHMARK DATA-SET

The UA-DETRAC benchmark data-set consists of 100 video sequences. The data-set is divided by 60 video sequences for train and 40 video sequences for test. The videos have 25 FPS and are taken at 24 different locations at Beijing and Tianjin in China including four categories of weather conditions(cloudy, night, sunny, and rainy), four categories of vehicle(car, bus, van, and others), large variation with scale and pose. The UA-DETRAC benchmark data-set contains 1.21million labeled bounding boxes of vehicles. The train and test video sequences comprise 83,791 and 56,340 frames with an resolution of 960 × 540 pixels respectively. Annotations, which include four categories of vehicle(car, bus, van and other), are only available for the train sequences. However, We can observed that 10 hard video sequences are significantly different from most train video sequences. Hard video sequences includes more complex traffic scenes that have large number of small and heavy occluded vehicles than easy and medium one.

B. TRAINING DETAILS

For vehicle detection tasks, we use MS-COCO [27] pre-trained Yolo-v3 model to initialize our backbone Darknet-53 network. The our proposed architecture with 608 × 608 resolution is trained in the end-to-end manner with Stochastic Gradient Descent (SGD) [28], where batch size is 64, subdivisions is 32, momentum is 0.9, weight decay is 0.0005, and on four NVIDIA GeForce GTX TITAN XP GPU with 12GB memory. We uses dual IOU thresholds and truth assignment similar as Faster R-CNN. If the IOU between a prediction and a ground truth bounding box is over 0.7, it is as a positive example. The learning rate is set of 0.001, and reduces from 10^{-3} to 10^{-5} by 10^{-1}. With each learning rate, we trained 40K, 5K, and 5K iterations respectively. We change the network resolution every 10 batches with fixed input image. The network selects from the following multiples of 32: 544, 572, 608, 640, 672 as similar manner of Yolo-v2 and Yolo-v3 [5], [22]. The network is resized by that dimension and continue training. The experiments were done with cuDNN v7.1 and CUDA 9.1.

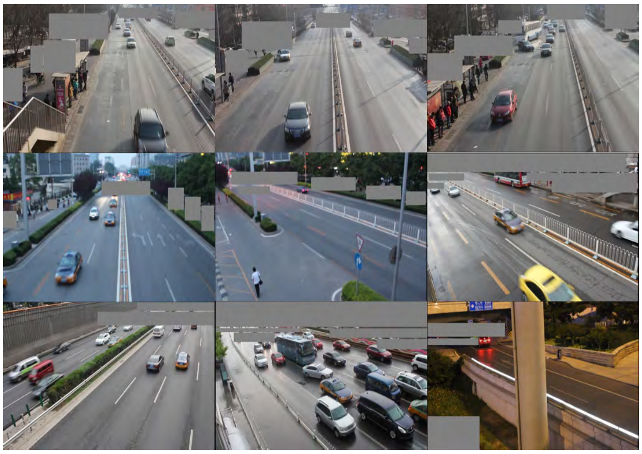


FIGURE 5. Sample images of the UA-DETRAC benchmark training-set that includes ignore regions.

C. EFFECT OF NUMBER OF SPP-NETWORK

To explore the effectiveness of SPP-network, the performance of our proposed 2 more prediction layers Yolo-v3 model trained by a different number of SPP-network is investigated. Table 2 illustrates the results which show the effectiveness of the different number of SPP-network. We can observe that as the number of SPP-networks increases, the overall mean average precision increases. One possible reason for this observation is that the SPP-network accepts variable sizes as input with multiple pooling layers to increase the robustness of the network performance. However, with the increase of the number of SPP-network, the run-time speed become slower. The proposed detector with 5 SPP-networks outperforms that by model with none SPP-network with gaps of 2.92% on the UA-DETRAC benchmark test-set.

D. DETECTION ACCURACY & SPEED

Table 3 and Figure 6 shows the results of our proposed architecture with the current state-of-the-art vehicle

TABLE 3. **Comparison to current leader of the UA-DETRAC detection challenge and to the top-performing detectors of the IWT4S Challenge on Advanced Traffic Monitoring 2017. Our proposed approach outperformed our baselines and all other approaches on the leader-board in the UA-DETRAC benchmark test-set suite.**

Method	Overall	Easy	Medium	Hard	Cloudy	Night	Rainy	Sunny
DPM	25.70	34.42	30.29	17.62	24.78	30.91	25.55	31.77
ACF	46.35	54.27	51.52	38.07	58.30	35.29	37.09	66.58
R-CNN	48.95	59.31	54.06	39.47	59.73	39.32	39.06	67.52
Fater R-CNN2	58.45	82.75	53.05	35.50	46.91	64.92	39.24	67.40
SA-FRCNN	45.83	73.93	49.00	30.76	49.97	52.30	33.39	55.04
NANO	63.01	80.33	68.04	50.73	67.00	62.20	55.89	73.89
CompACT	53.23	64.84	58.70	43.16	63.23	46.37	44.21	71.16
EB	67.96	89.65	73.12	53.64	72.42	73.93	53.40	83.73
R-FCN	69.87	93.32	75.67	54.31	74.38	75.09	56.21	84.08
GP-FRCNN	77.96	92.74	82.39	67.22	83.23	77.75	70.17	86.56
HAVD	80.51	94.48	86.13	69.02	87.28	82.30	69.37	89.71
SSD-VDIG	82.68	94.60	**89.71**	70.65	**89.81**	83.02	73.35	88.11
Our Proposed Method	**85.29**	**96.04**	89.42	**76.55**	88.00	**88.67**	**78.90**	**88.91**

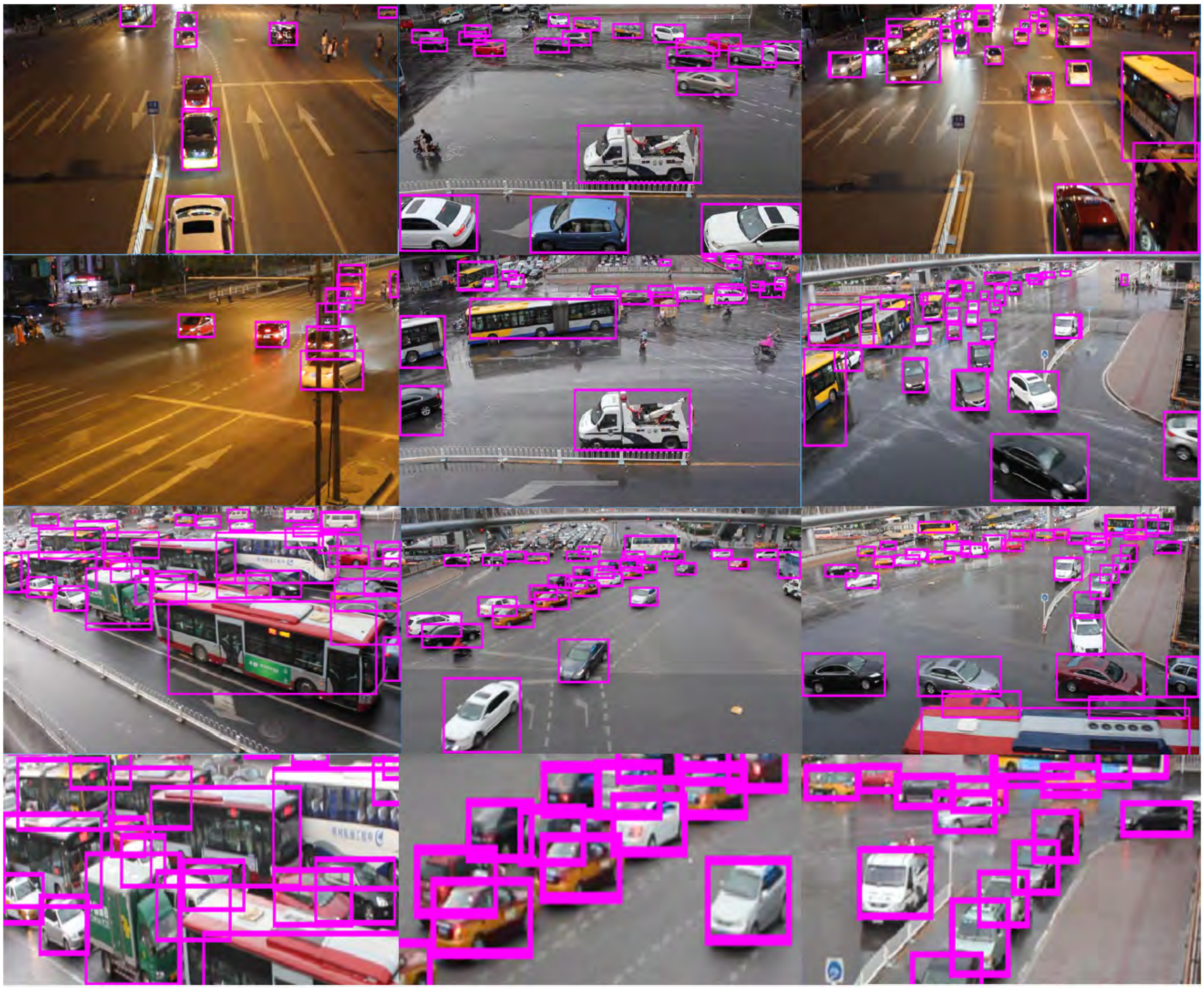


FIGURE 6. **Qualitative detection results of out proposed method on the UA-DETRAC benchmark test-set. The result bounding boxes are dense around heavy occluded vehicles and correctly detected for hard traffic scene.**

detection approaches of the 2018 UA-DETRAC detection challenge track 2, the winner detectors of the IWT4S challenge on Advanced Traffic Monitoring 2017 [29] and baseline detectors. From Table 3, we can see that the overall mean average precision of our proposed architecture is 85.29 on UA-DETRAC benchmark test-set,

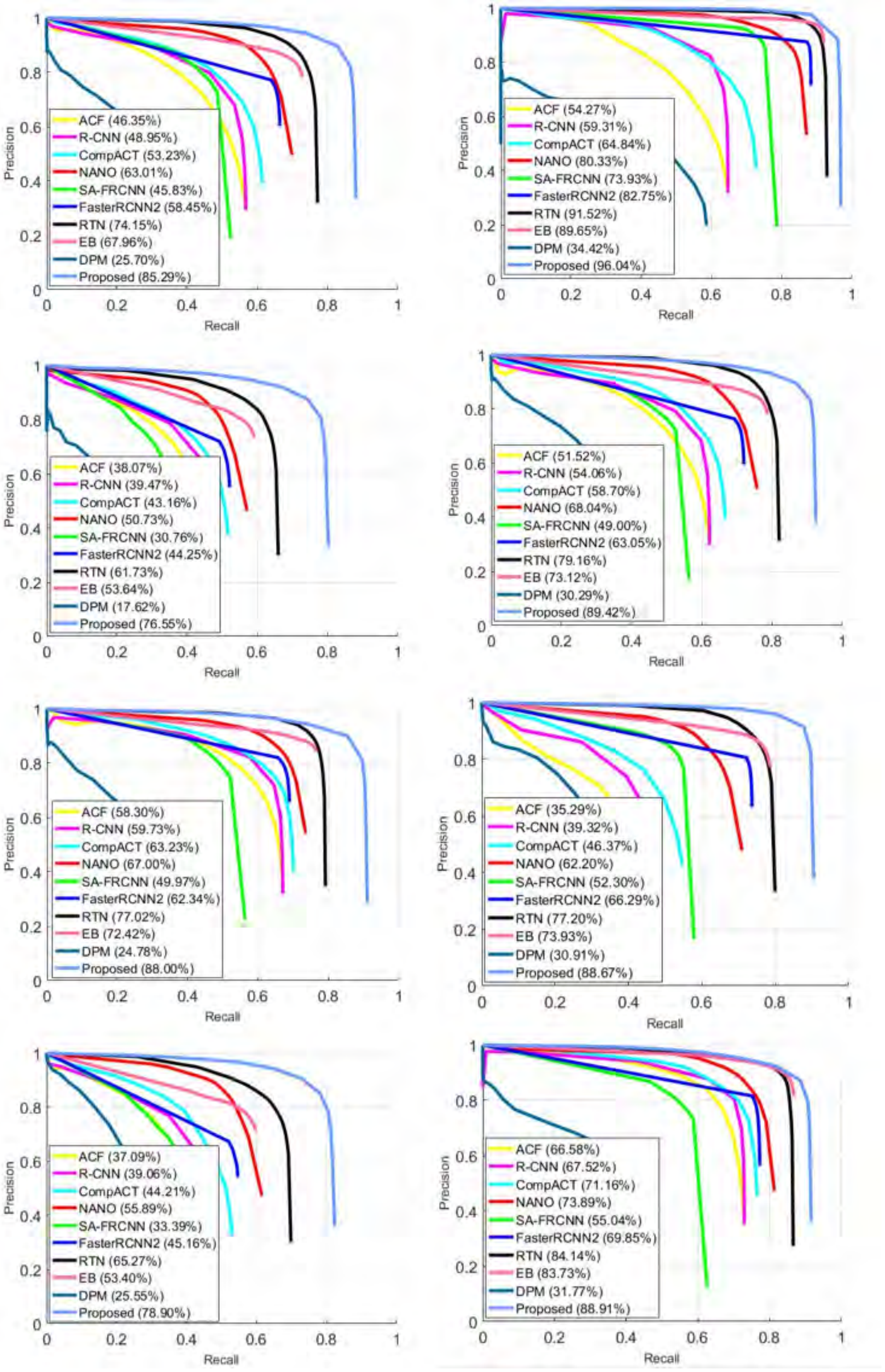


FIGURE 7. Precision-recall curves of different vehicle detection algorithms on the full UA-DETRAC benchmark test-set. Clockwise from top left: Overall; easy; medium; hard; cloudy; night; rainy; sunny sequences of the UA-DETRAC bench-mark test-sets.

which is 59.59%, 38.94%, 36.34%, 26.84%, 39.46%, 22.28%, 32.06%, 17.33%, 15.42%, 7.33%, 4.78%,and 2.61% higher than that of, DPM, ACF, R-CNN, Faster R-CNN2, SA-FRCNN, NANO, CompACT [30], EB [31], R-FCN, GP-FRCNN [32], HAVD, SSD-VDIG respectively. Proposed approach clearly outperforms all other detectors on the full test-set(overall). Compared to GP-FRCNN which is the winner detector of 2017 Challenge track 2, Proposed detector is improved by 7.33 percentages. As mention it before, the UA-DETRAC benchmark data-set consists of three levels of difficulty (easy, medium, hard). We clearly improve the mAP for hard sequences. Figure 7 shows precision-recall curves of object detection methods on overall, easy, medium, hard, cloudy, night, rainy, sunny sequences of the UA-DETRAC benchmark test-sets. The gap between ours and baseline methods was not significant on easy and sunny sequences, but it was more noticeable on hard and rainy sequences. The Speed was measured by using the forward path of the network with a batch size 1. The run-time speed is 9-10 FPS on a single NVIDIA GeForce GTX TITAN XP GPU. The conventional Yolo-v3-608 model has 20-25 FPS. Our model is about 3 times slower than the conventional one, slightly slower than the Faster R-CNN2 (11.11 fps) but faster than CompACT (0.22 fps), ACF (0.67 fps) and DPM (0.17 fps). The current top-performing detectors(SSD-VDIG, HAVD) of the UA-DETRAC challenge are more than 2 times slower than ours.

E. QUALITATIVE RESULTS

Figure 6 depicts qualitative evaluations of our approach on the UA-DETRAC benchmark test-set. We successfully detect most of the vehicles in different appearances, especially when heavy and partial occlusions are occurred, also the vehicles that are far away from the camera for challenging scene such as high traffic density and various lightning conditions. We zoomed in detection results (Three images at the bottom) to verify that even heavily occluded vehicles are correctly detected by our proposed detector.

V. CONCLUSION

In this paper, we proposed a multi-scale vehicle detection with spatial pyramid pooling method when improves the conventional Yolo-v3 for robust detection to the scale change of the vehicle and the occlusion. The contribution of the paper is as follows. First, it adds two more object prediction layers based on the conventional Yolo-v3 model to detect vehicles effectively in different scales. One additional prediction layer creates between lager size object perdition layer and medium. Another additional prediction layer creates between medium size object prediction layer and small. Second, the SPP-networks were implemented before each prediction layer after feature pyramid network to improve accuracy by increasing the number of features without much time overhead. The our proposed architecture shows a state-of-the-art mAP detection ratio against the others vehicle detection approaches with reasonable run-time speed (9-10 FPS).

REFERENCES

[1] B.-G. Han, J. T. Lee, K.-T. Lim, and Y. Chung, "Real-time license plate detection in high-resolution videos using fastest available cascade classifier and core patterns," *ETRI J.*, vol. 37, no. 2, pp. 251–261, 2015.

[2] K.-J. Kim, P.-K. Kim, K.-T. Lim, Y.-S. Chung, Y.-J. Song, S. I. Lee, and D.-H. Choi, "Vehicle color recognition via representative color region extraction and convolutional neural network," in *Proc. 10th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2018, pp. 89–94.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 21–37.

[5] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767

[6] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. CVPR*, Jul. 2017, vol. 1, no. 2, pp. 2117–2125.

[7] K.-J. Kim, P.-K. Kim, Y.-S. Chung, and D.-H. Choi, "Performance enhancement of YOLOv3 by adding prediction layers with spatial pyramid pooling for vehicle detection," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Nov. 2018, pp. 1–6.

[8] 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.

[9] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," 2015, *arXiv:1511.04136*. [Online]. Available: https://arxiv.org/abs/1511.04136

[10] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2241–2248.

[11] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[13] G. Cheng, P. Zhou, and J. Han, "RIFD-CNN: Rotation-invariant and Fisher discriminative convolutional neural networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2884–2893.

[14] G. Cheng, J. Han, P. Zhou, and D. Xu, "Learning rotation-invariant and Fisher discriminative convolutional neural networks for object detection," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 265–278, Jan. 2019.

[15] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.

[16] 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.

[17] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[18] 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, Apr. 2013.

[19] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2014, pp. 391–405.

[20] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," 2013, *arXiv:1312.6229*. [Online]. Available: https://arxiv.org/abs/1312.6229

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.

[22] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," 2017, *arXiv:1612.08242*. [Online]. Available: https://arxiv.org/abs/1612.08242

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[25] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5562–5570.

[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[28] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*. Springer, 2010, pp. 177–186.

[29] S. Lyu *et al.*, "UA-DETRAC 2017: Report of AVSS2017 & IWT4S challenge on advanced traffic monitoring," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug./Sep. 2017, pp. 1–7.

[30] Z. Cai, M. Saberian, and N. Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3361–3369.

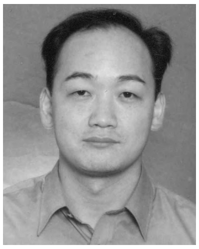
[31] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue, "Evolving boxes for fast vehicle detection," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 1135–1140.

[32] S. Amin and F. Galasso, "Geometric proposals for faster R-CNN," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug./Sep. 2017, pp. 1–6.

KWANG-JU KIM received the B.S. degree in electronics engineering from Kyungpook National University (KNU), Daegu, South Korea, in 2010, and the M.S. degree in electrical engineering from the Pohang Institute of Science and Technology (POSTECH), Pohang, South Korea, in 2013. From 2013 to 2015, he was a Researcher with General Electric. Since 2015, he has been with the Electronics and Telecommunications Research Institute, Daejeon, South Korea. His current research interests include computer vision, pattern recognition, and video surveillance.

PYONG-KUN KIM was born in Seoul, South Korea, in 1974. He received the B.S. and M.S. degrees in electrical engineering from Seoul National University, in 1997 and 1999, respectively.

From 1999 to 2001, he was a Researcher with Sindo-Ricoh. Since 2002, he has been a Researcher with the Electronics and Telecommunication Research Institute. He holds two patents about license plate recognition. His research interests include parameter estimation, automatic letter sorting machine, postal address processing, machine learning, license plate recognition, optical character recognition, object detection, and synthetic data generation.



YUN-SU CHUNG received the M.S. and Ph.D. degrees in electronics engineering from the Department of Electronics Engineering, Kyungpook National University, Daegu, South Korea, in 1995 and 1998, respectively. Since 1999, he has been with the Electronics and Telecommunications Research Institute, Daejeon, South Korea. He is currently the Leader of the Regional Industry IT Convergence Research Section. His current research interests include bio-metrics, video surveillance, human–robot interface, and human–computer interface.

DOO-HYUN CHOI received the B.S. degree in electronics engineering from Kyungpook National University (KNU), Daegu, South Korea, in 1991, and the M.S. and Ph.D. degrees in computer science and engineering from the Pohang Institute of Science and Technology (POSTECH), Pohang, South Korea, in 1993 and 1996, respectively. From 1996 to 2000 and from 2000 to 2003, he served as an Assistant Professor with KNU and Seoul National University, respectively. In 2003, he rejoined KNU and became a Full Professor with the School of Electronics Engineering. Also, he serves as the Head of the Department of Mobile Engineering supported by Samsung Electronics and is in charge of the Intelligent Information Systems Lab, KNU. His research interests include intelligent information and signal processing systems, and the practical development capacity building of future generations.