MULTI-SCALE FEATURE FUSION AND MULTI-LAYER PROPOSAL NETWORK FOR TWO-STAGE OBJECT DETECTION IN AUTONOMOUS DRIVING SYSTEMS

HOANH NGUYEN

Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

E-mail: nguyenhoanh@iuh.edu.vn

ABSTRACT

In recent years, many deep learning-based approaches for object detection in autonomous driving systems have been proposed and achieved great achievements. These approaches can be divided into two groups: one-stage methods and two-stage methods. Compared with one-stage methods, two-stage methods achieve better detection performance. However, the performance of two-stage methods is still limited with objects at different scales and heavy occluded objects because there is little discriminative feature to exploit in the last convolution layer of the base network in two-stage architecture. To solve this problem, this paper proposes an improved two-stage framework based on Faster R-CNN architecture for object detection in autonomous driving systems. In the proposed framework, the base network based on VGG16 architecture is first adopted to generate the base convolution layers. To increase the performance of detecting objects at different scales, multi-feature concatenation modules are used at different convolution blocks of the base network. The proposed multi-feature concatenation modules combine all sub-layers of each convolution block to generate enhanced feature maps, which contain more discriminative features. All enhanced feature maps generated by multi-feature concatenation modules are then fed to an improved multi-layer region proposal network module. Each improved RPN contains a 1×1 convolution layer for compressing the input channel and a 3×3 dilated convolution layer for increasing the receptive field. To solve the issue of heavy duplicate proposals in traffic scene images, soft-NMS algorithm is adopted to keep final proposals. Finally, all good proposals are fed to a detection sub-network which includes a RoI pooling layer and fully connected layers for classifying objects and regressing the coordinates of each detected object. Experimental results on Pascal VOC dataset and KITTI dataset show that the proposed method outperforms Faster R-CNN in detection accuracy.

Keywords: *Object Detection, Convolutional Neural Network, Autonomous Driving Systems, Deep Learning, Region Proposal Network*

1. INTRODUCTION

Vision-based object detection is an important research problem for computer vision, with a wide range of real-world applications such as robotic applications, surveillance, drone technology, autonomous driving systems and so on. The main task of vision-based object detection is to predict the position and category of objects from images or videos. Compared to image classification, there are many more aspects that object detection needs to address. As a result, the computational complexity is significantly higher than that of the image classification. In addition, most applications need to perform object detection in real-time. Thus, the inference speed is an important concern in object detection aspect. Vision-based object detection methods can be divided into two groups: traditional methods and deep learning-based methods. In traditional methods, hand-crafted features have been used to detect multiple classes of objects. Traditional methods achieved good performance in limited environment conditions. In complicated environments such as traffic scenes, the performance of traditional methods is greatly reduced. Since AlexNet achieved large success in the ImageNet challenge [29], deep learning quickly becomes the dominant object detection approach. Although deep learning-based object detectors achieved great achievements in recent years, real-time visual object detection in autonomous driving environment is still very challenging. It is very harsh for visual object detection with poor illumination and weather conditions in autonomous driving environment. In addition, there can be many occluded and truncated objects with large object scale variations in traffic scenes. It is observed that the object detection performance of the popular deep learning detectors such as Faster-RCNN and SSD without modification is not very good. Apart from the detection accuracy, the inference speed is also a large concern for autonomous driving systems. Vehicles are unlikely to be equipped with GPU computers as powerful as used in research environments.

Based on above research problems, this paper proposes an improved two-stage framework based on Faster R-CNN architecture for object detection in autonomous driving systems. In the proposed framework, the base network based on VGG16 architecture is first adopted to generate the base convolution layers. To increase the performance of detecting objects at different scales, multi-feature concatenation modules are used at different convolution blocks of the base network. The proposed multi-feature concatenation modules combine all sub-layers of each convolution block to generate enhanced feature maps, which contain more discriminative features. All enhanced feature maps generated by multi-feature concatenation modules are then fed to an improved multi-layer RPN module. The improved RPN module is designed based on original RPN with modification to increase the inference speed. Each improved RPN contains a 1×1 convolution layer for compressing the input channel and a 3×3 dilated convolution layer for increasing the receptive field. Since anchors generated by the multi-layer RPN module are usually overlap, proposals end up also overlapping over the same object. To solve the issue of heavy duplicate proposals, Soft-NMS algorithm is adopted to keep final proposals. Finally, all good proposals are fed to a detection sub-network which includes a RoI pooling layer and fully connected layers for classifying objects and regressing the coordinates of each detected object.

This paper is organized as follows: an overview of previous methods is presented in Section 2. Section 3 describes detail the proposed method. Section 4 demonstrates experimental results. Finally, the conclusion is made in Section 5.

2. RELATED WORK

2.1 Object Detection Method

Object detection methods can be divided into two groups: traditional methods and deep learning-based methods. Traditional methods usually include three major stages: finding object proposals, extracting features, and classification. For finding object proposals stage, traditional approaches used selective search [13], edge box [16], multiscale combinatorial grouping [17], and so on. For extracting features stage, histogram of oriented gradient [18], scale-invariant feature transform [19], and so on are some of the methods usually used in traditional approaches. For classification stage, support vector machine [20], adaboost [21], and so on are used in traditional approaches. Traditional methods achieved good performance in limited environment conditions. In complicated environments such as traffic scenes, the performance of traditional methods is greatly reduced.

With the fast development of deep learning, many deep learning-based studies for object detection in autonomous driving systems have been proposed and achieved successful results. These deep learning-based approaches can be divided into two groups: one-stage approaches and two-stage approaches. For one-stage architecture, the input images are sliced into several grid cells. The classifier outputs a vector that encodes the information of each grid cell. Compared with two-stage approaches, one-stage approaches are faster and easier to train while yielding inferior performance [25]. Popular representatives of one-stage approaches include YOLOv2 [22], YOLOv3 [23], SSD [12], DSSD [15]. SSD skips the region proposal stage and directly uses multiple feature maps with different resolutions to perform object localization and classification. YOLO and YOLOv2 are other one-stage detectors that can achieve even faster speed at the expense of accuracy. By introducing improvements of batch normalization, high resolution classifier, convolutional with anchor boxes and dimension clusters to YOLO, YOLOv2 achieves higher accuracy and higher speed. DSSD proposed to augment SSD+Residual-101 with deconvolution layers to introduce additional large-scale context in object detection and improve accuracy, especially for small objects.

The two-stage approaches first find the ROIs and then performs detection in every ROI. Comparing the two approaches, the one-stage approach predicts the classes and locations of objects directly, while the two-stage approach finds ROI first and then performs the classifications on an ROI. Popular representatives of two-stage approaches include Fast R-CNN [24], Faster R-CNN [1], FPN

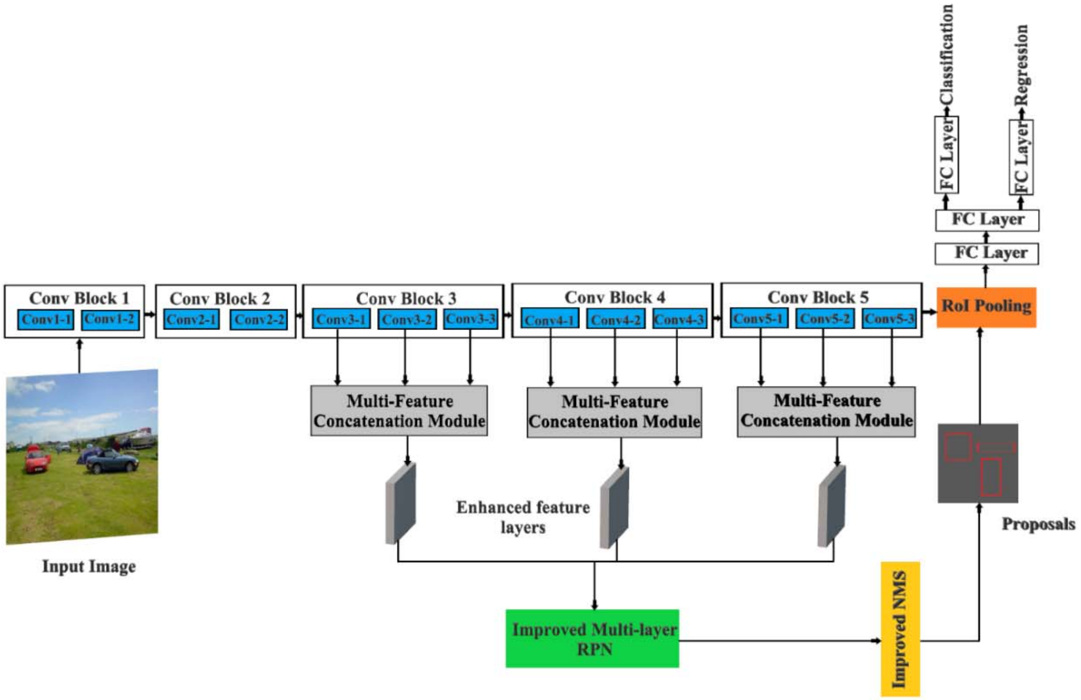

Figure 1: The Overall Architecture of The Proposed Framework.

[11], R-FCN [14]. R-FCN proposed region-based fully convolution network based on positive-sensitive cropping to reduce the number of ROIs per image. R-FCN achieved comparable accuracy with a speed that was slighter higher than that of ResNet-101. FPN exploited the inherent multi-scale, pyramidal hierarchy of deep convolutional networks to construct feature pyramids with marginal extra cost. A top-down architecture with lateral connections is developed for building high-level semantic feature maps at all scales.

2.2 Object Detection Based on R-CNN

In the line of two-stage deep learning-based object detectors, R-CNN [26] is a pioneer deep learning model, which increases object detection accuracy over traditional detectors by a large margin. In the first stage, R-CNN applies selective search method [13] to generate sufficient proposal candidates that contain all the objects. In the second stage, R-CNN forwards each proposal through convolutional networks, followed by classifying the proposals with SVMs [20] and predicting bounding boxes offsets with linear regression. However, this method is very time-consuming, as every proposal is processed by the entire network. Fast R-CNN [24] extends R-CNN by using one single convolution network to perform shared computation in the second stage, which increases the speed significantly. The problem with Fast R-CNN is that the proposals are generated by a traditional time-consuming selective search algorithm. Faster R-CNN [1] was proposed to further improve upon Fast R-CNN. Faster-RCNN proposed region proposal network (RPN) to replace selective search method in R-CNN and makes the whole network trainable in an end to end approach.

Recently, several approaches have been proposed to increase the accuracy of Faster R-CNN. Instead of using VGG-16 architecture as a base network for Faster R-CNN, adoption of different backbone networks, such as ResNet and Inception ResNet, has been proposed. He et al. [4] proposed the use of a deep residual network, such as ResNet-101, for image recognition. The authors showed that ResNet-101 has a lower complexity compared to VGG-16 and achieves good accuracy. Lin et al. [11] proposed using a feature pyramid network (FPN) for Faster-RCNN. With feature sharing, the FPN-based Faster R-CNN system achieved better accuracy than original Faster R-CNN. Huang et al. [25] used an Inception ResNet v2 in the backbone of the Faster R-CNN to achieve better accuracy than that obtained using ResNet 101 with a slightly lower running time per frame. Shrivastava et al. [27] proposed a top–down modulation (TDM) network to incorporate fine details in the detection network for detecting small objects. They achieved higher accuracy compared to [25] with a slightly higher frame rate. Yauan et al. [28] proposed two refinement methods,

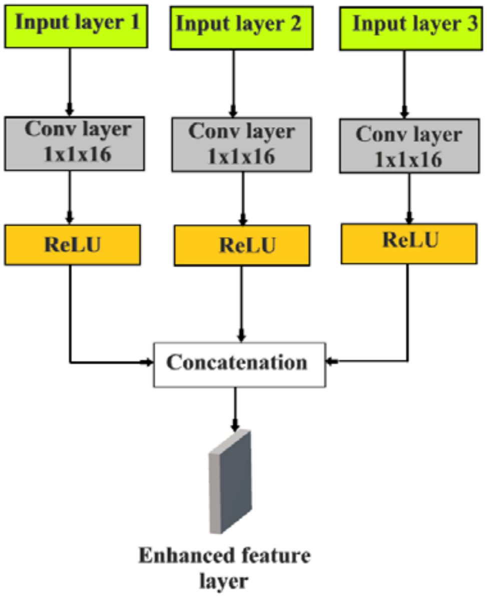

Figure 2: The Architecture of The Multi-Feature Concatenation Module.

iterative and LSTM refinement, for the Faster R-CNN model and improved the accuracy.

3. PROPOSED APPROACH

Figure 1 shows the overall architecture of the proposed approach. As shown in Figure 1, the base network based on VGG16 architecture [8] is first adopted to generate the base convolution layers. To increase the performance of detecting objects at different scales, multi-feature concatenation modules are used at convolution block 3, block 4 and block 5 of the base network. The proposed multi-feature concatenation modules combine all sub-layers of each convolution block to generate enhanced feature maps, which contain more discriminative features. All enhanced feature maps generated by multi-feature concatenation modules are then fed to an improved multi-layer region proposal network (RPN) module. Each improved RPN contains a 1×1 convolution layer for compressing the input channel and a 3×3 dilated convolution layer for increasing the receptive field. Since anchors generated by the multi-layer RPN module are usually overlap, proposals end up also overlapping over the same object. To solve the issue of heavy duplicate proposals, Soft-NMS algorithm is adopted to keep final proposals. Finally, all good proposals are fed to a detection sub-network which includes a RoI pooling layer and fully connected layers for classifying objects and regressing the coordinates of each detected object. Details of each proposed module will be explained in the following sections.

3.1 Enhanced Feature Map Generation by Multi-Feature Concatenation

In two-stage object detection frameworks such as Faster R-CNN [1], the last convolution layer of the base network is adopted to generate object-like regions by the region proposal network. However, single-layer convolutional feature maps often lack some information of original image, thus decreasing the detection performance of these frameworks. To solve this problem, this paper proposes to generate enhanced feature layers from original layers of the base network by using multi-feature concatenation module as proposed in [2]. Figure 2 shows the architecture of the multi-feature concatenation module. From Figure 1 and Figure 2, it can be observed that the multi-feature concatenation module is applied on the Conv3, Conv4 and Conv5 blocks of the VGG16 network. In each multi-feature concatenation module, each sub-layer Cij ($i = 3, 4, 5$ denotes the block and $j = 1, 2, 3$ denotes the sub-layers in this block) of the convolution block is used. Although the last layer of each convolution block has more discriminative features, the combination of all feature maps will enhance more dominant features, thus increasing the detection performance of the network. However, directly concatenate all feature maps will lead to a large number of output channels of output feature maps, which lead to reduce the inference speed and computational issues. Thus, a 1×1×16 convolution layer is used after each

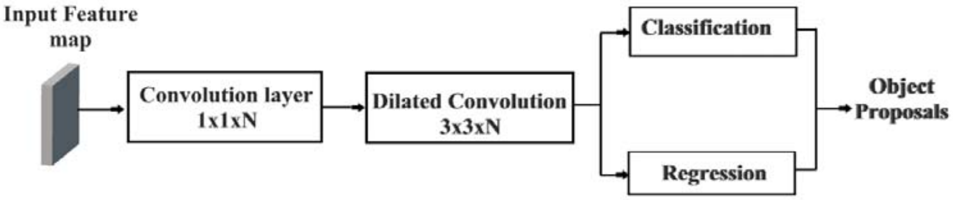

Figure 3: The Architecture of Improved Region Proposal Network.

sub-layer to decrease the number of output channels to 16 channels. Then, all sub-layers in each convolution block are concatenated to generate the enhanced feature layers.

3.2 Proposal Generation with Improved Region Proposal Network

The Region Proposal Network (RPN) is first introduced in Faster R-CNN [1]. The RPN first generates a set of anchor boxes at each location of the last convolution layer of the base network. Then, the RPN classifies these anchor boxes to object/background class and regresses the coordinates of these anchor boxes. There are 9 anchor boxes in total at each location of the feature map in original Faster R-CNN framework. Each anchor box is associated with predefined scales and aspect ratios. This paper uses multi-layer RPNs on each enhanced feature layers for generating object proposals. To increase the inference speed and detection accuracy, an improved region proposal network is designed as shown in Figure 3. First, to improve the inference speed of multi-layer RPNs on different enhanced feature map layers, this paper reduces the number of channels of the input feature layers to decrease the number of parameters in the subsequent convolutional layer. Recently, 1×1 convolution layer is usually used to reduce the number of parameters without losing accuracy while also gaining efficiency [3] [4]. Thus, this paper uses 1×1 convolution layer with 64 channels to make the architecture simpler. Next, the 3×3 convolution layer as in original RPN is replaced by the dilated convolution. Dilated convolution is usually used in the context of semantic segmentation [5] [6]. Dilated convolution can increase the receptive field, thus effectively enlarging the field of view of filters to incorporate larger context without increasing the number of parameters or the amount of computation. With larger receptive field, the network can see a bigger context information and recognize more confined bounding boxes. To further increase the accuracy of localization of bounding boxes, the continuous dilated convolution as in [5] is adopted in this paper. In the continuous dilated convolution, dilated kernels are applied in the previous convolutional layers.

3.3 Improved Non-Maximum Suppression Algorithm

The multi-layer RPN generates a large number of region proposals, and each region proposal has a corresponding score, and adjacent region proposals have relevant scores, which may cause false detection results and may result in some overlapping objects are missed. To solve this problem, non-maximum suppression algorithm (NMS) is adopted in most state-of-the-art object detection frameworks, including Faster R-CNN.

Non-Maximum Suppression Algorithm

Let $P_{in} = \{p_1, p_2, p_3, \dots, p_n\}$ denotes the initial proposals generated by the multi-layer RPN, in which the proposals are sorted by their objectiveness scores. The objectiveness score S_i for proposal p_i is the maximum value in the classification score vector of p_i. For a proposal p_i, any other proposal that has an overlap more than a pre-defined threshold T with this proposal is called a neighbor proposal of proposal p_i. The traditional NMS algorithm works as the flowchart shown in Figure 4.

Traditional NMS removes any other proposal that has an overlap more than a pre-defined threshold with a winning proposal. However, due to heavy object occlusion in real-life environments, traditional NMS algorithm may remove positive proposals unexpectedly. To address this issue, this paper adopts soft-NMS algorithm [7]. With soft-NMS, the neighbor proposals of a winning proposal are not completely suppressed. Instead they are suppressed based on the updated objectiveness scores of the neighbor proposals, which are computed according to the overlap level of the neighbor proposals and the winning proposal.

Soft-NMS Algorithm

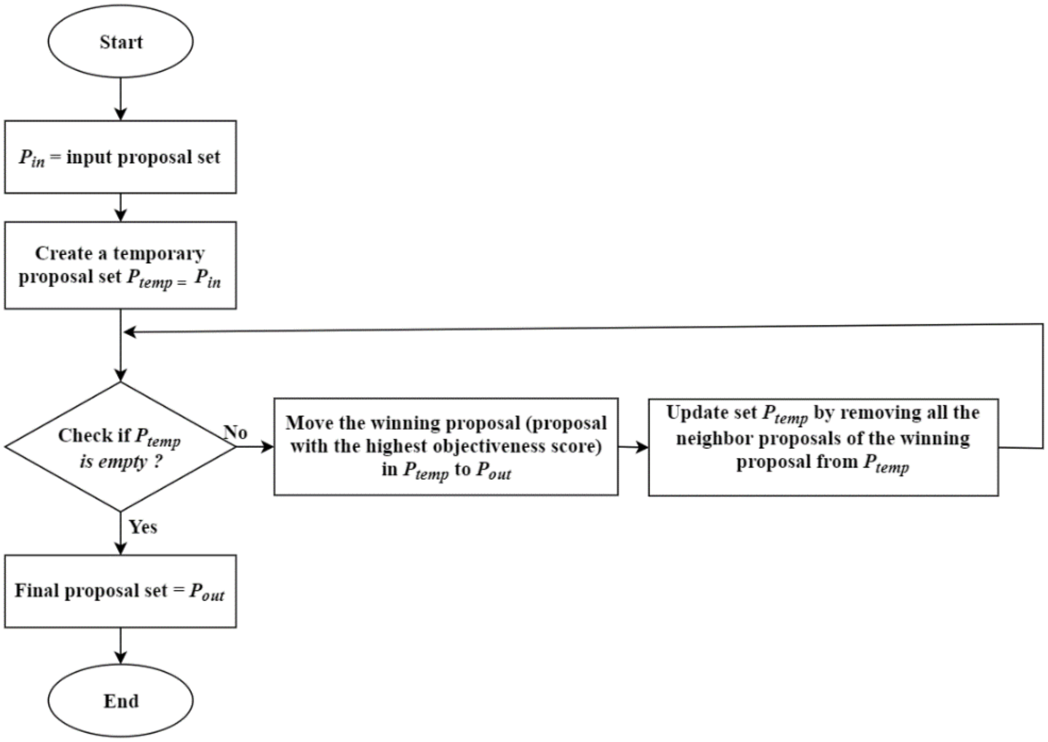
Start
P_{in} = input proposal set
Create a temporary proposal set $P_{temp} = P_{in}$
Check if P_{temp} is empty ?
No
Move the winning proposal (proposal with the highest objectiveness score) in P_{temp} to P_{out}
Update set P_{temp} by removing all the neighbor proposals of the winning proposal from P_{temp}
Yes
Final proposal set = P_{out}
End

Figure 4: The Flowchart of The Non-Maximum Suppression Algorithm.

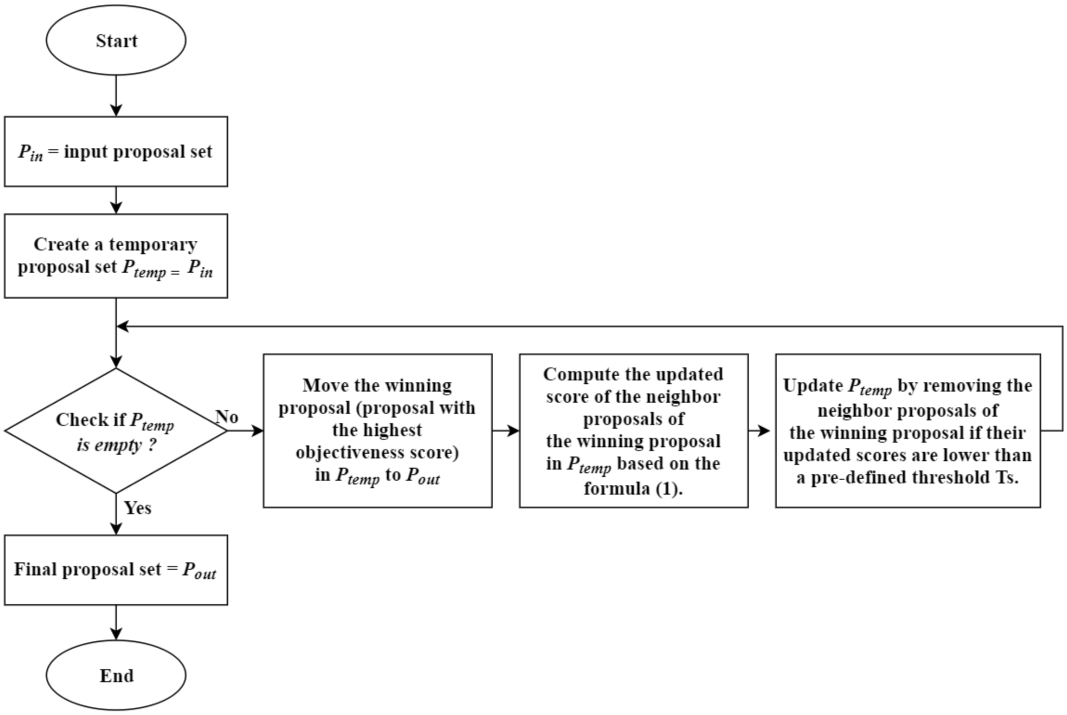
Start
P_{in} = input proposal set
Create a temporary proposal set $P_{temp} = P_{in}$
Check if P_{temp} is empty ?
No
Move the winning proposal (proposal with the highest objectiveness score) in P_{temp} to P_{out}
Compute the updated score of the neighbor proposals of the winning proposal in P_{temp} based on the formula (1).
Update P_{temp} by removing the neighbor proposals of the winning proposal if their updated scores are lower than a pre-defined threshold Ts.
Yes
Final proposal set = P_{out}
End

Figure 5: The Flowchart of The Soft-Non-Maximum Suppression Algorithm.

Let S'_j denotes the updated objectiveness scores of the neighbor proposal p_j of the winning proposal p_i. S'_j is computed with the following formula [7]:

$$S'_j = S_i(1 - I_{p_i,p_j}) \tag{1}$$

where I_{p_i,p_j} denotes the intersection of union (IoU) between proposal p_i and proposal p_j and is computed by the following formula:

$$I_{p_i,p_j} = \frac{area(p_i \cap p_j)}{area(p_i \cup p_j)} \tag{2}$$

Soft-NMS algorithm works as the flowchart shown in Figure 5.

3.4 Subnetwork of Detection with RoI-Pooling Layer and Fully Connected Layer

The detection subnetwork includes RoI pooling layer and fully connected (FC) layer. The RoI pooling layer uses max pooling operation to convert the features inside any valid RoI into a small feature map with a fixed spatial extent of H × W. RoI max pooling works by dividing the h × w RoI proposal into an H × W grid of sub-windows of approximate size h/H × w/W, and then max-pooling the values in each sub-window into the corresponding output grid cell. If a proposal is smaller than H×W, it will be enlarged to H × W by adding replicated values to fill new space. RoI pooling avoids repeatedly computing the convolutional layers, so it can significantly speed up both train and test time. However, the max pooling operation brings about the problem of misalignment obviously. For designing a fast and efficient detection framework, this paper performs the RoI pooling by cropping a feature region generated by the multi-layer RPN and resizing the region to the fixed spatial extent of 14 × 14 via bilinear interpolation. Subsequently, the fixed size feature map is fed into two fully connected layers sequentially, and subsequently another two sibling fully connected layers for classification and localization. The first FC layer is fed into the softmax layer to compute the confidence probabilities of being objects and background. The second FC layer with linear activation functions regresses the bounding box of detected objects.

4. EXPERIMENTAL RESULTS

4.1 Dataset

In order to compare the effectiveness of the proposed approach with other state-of-the-art approaches, this paper conducts experiments on the Pascal VOC dataset [9] and KITTI dataset [10]. Pascal VOC dataset is a widely used dataset for evaluating object detection frameworks. Pascal VOC dataset contains 20 categories of indoor and outdoor objects class. This paper mainly focuses on the average precision (AP) of six classes appearing in traffic scenes, including car, person, bus, bike, motorbike, train, and the mean average precision (mAP) of the whole dataset. Following the instructions as in [9], this paper uses the mAP@0.5 metric to evaluate accuracy. This paper adopts VOC07+12 dataset, which contains 16551 images for training and 4952 images for testing to evaluate the proposed approach.

KITTI dataset [10] is a large dataset for evaluating object detection approaches in driving environments. KITTI dataset contains 7481 images for training and 7518 for testing. The image size is 384×1280 pixels. Each image in this dataset includes two classes: car and pedestrian. Since no ground truth is available for the test set, this paper splits the training set into training set and validation set by 8:1.

4.2 Implementation Details

The proposed approach is implemented in Pytorch deep-learning framework with Python interface. The CPU used in all experiments is Intel Core i7-8700, the main memory is 12GB DDR4 RAM, and the GPU is NVIDIA GeForce GTX 1080. In the training phase, this paper sets the training iteration as 50k. A learning rate of 0.001 for the first 30k iterations and 0.0001 for the remaining iterations is adopted. Following the original paper [1], the input image is resized, such that the length of the shorter side of the image is 600 pixels. Horizontal flipping is used for data augmentation.

4.3 Detection Results on Pascal VOC Dataset

In this section, this paper conducts experiments on the Pascal VOC dataset and compares the detection results with the results of other state-of-the-art approaches, including FPN [11], SSD [12], R-FCN [14], DSSD [15], and Faster R-CNN [1]. DSSD proposed to augment SSD+Residual-101 with deconvolution layers to introduce additional large-scale context in object detection and improve accuracy, especially for small objects. R-FCN presented region-based, fully convolutional networks for accurate and efficient object detection. SSD presented a method for detecting objects in images using a single deep neural network. FPN exploited the inherent multi-scale, pyramidal hierarchy of deep convolutional networks to construct feature pyramids with marginal extra cost.

Table 1: Comparison Detection Results with Other Approaches on Pascal VOC Dataset.

Method	Person (%)	Bike (%)	Car (%)	Bus (%)	Motorbike (%)	Train (%)	mAP (%)
FPN [11]	84.7	85.5	88.2	86.9	85.5	87.2	81.1
SSD [12]	83	87.6	88.1	88.2	87.5	87.2	80.6
R-FCN [14]	81.2	87.2	88.5	86.8	79.9	85.9	80.5
DSSD [15]	83.7	86.2	88.7	89.0	87.5	85.7	81.5
Faster R-CNN [1]	75.4	80.7	85.3	85.1	80.9	85.3	76.4
Proposed Method	84.9	85.8	89.0	87.1	86.0	86.2	80.4

Table 2: Detection Results on KITTI Dataset.

Method	Car (%)	Pedestrian (%)	mAP (%)
SSD	70.0	18.5	44.25
Faster R-CNN	82.1	68.5	75.3
Proposed Method	84.6	79.1	81.9

Table 1 shows the comparison of detection results on Pascal VOC dataset. As shown in Table 1, the proposed approach achieves the best AP result with person and car subsets. More specific, compared with the best result among reference methods, the AP of the proposed method is improved by 0.2% and 0.3% with person and car subset respectively. In terms of the mAP, the proposed method achieves comparable result compared with other state-of-the-art methods. From Table 1, DSSD achieves the best mAP result. DSSD adopts Residual-101 architecture as the base network and deconvolution layers to improve small object detection. Residual-101 architecture performs better performance compared to VGG16 architecture. However, VGG16 is shallower and simpler than Residual-101, thus enhancing the inference speed. In addition, compared with original Faster R-CNN framework, the proposed approach outperforms in both AP and mAP results. More specific, the mAP of the proposed method is improved by 4% compared with Faster R-CNN. Figure 6 shows some examples of detection results of the proposed method on Pascal VOC dataset. As shown in Figure 6, the proposed approach can exactly locate objects at different scales in difficult driving environments.

4.4 Detection Results on KITTI Dataset

To further evaluate the performance of the proposed method, this paper conducts experiments on the KITTI dataset and compares the detection results with the results of SSD and Faster R-CNN framework. Table 2 shows the detection results of all methods on the KITTI dataset. Compared with SSD and Faster R-CNN, the proposed method achieves the best detection results in both AP and mAP. More specific, the mAP of the proposed approach is improved by 37.65% and 6.6% compared with SSD and Faster R-CNN respectively. These results demonstrate the effectiveness of the proposed approach on detecting objects at different scales in difficult driving environments. Figure 7 shows examples of detection results on KITTI dataset of the proposed method (left column) and original Faster R-CNN framework (right column). As shown in Figure 7, the proposed method can detect objects in difficult conditions while Faster R-CNN misses some objects.

5. CONCLUSIONS

Vision-based object detection is one of the most critical problems for autonomous driving systems. In recent years, deep learning-based approaches achieved huge successes on visual object detection over traditional approaches. However, due to the difficult conditions in driving environments such as large object scale variation, object occlusion and so on, popular deep learning-based object detectors such as Faster-RCNN do not produce good detection performance. In this paper, an improved two-stage framework based on Faster R-CNN architecture for object detection in autonomous driving systems is proposed. Firstly, the base network based on VGG16 architecture is used to generate the base feature maps. Secondly, multi-feature concatenation modules are used at different convolution blocks of the base network to increase the performance of detecting objects at different scales. All enhanced feature maps generated by multi-feature concatenation modules are then fed to an improved multi-layer region proposal network module. Each

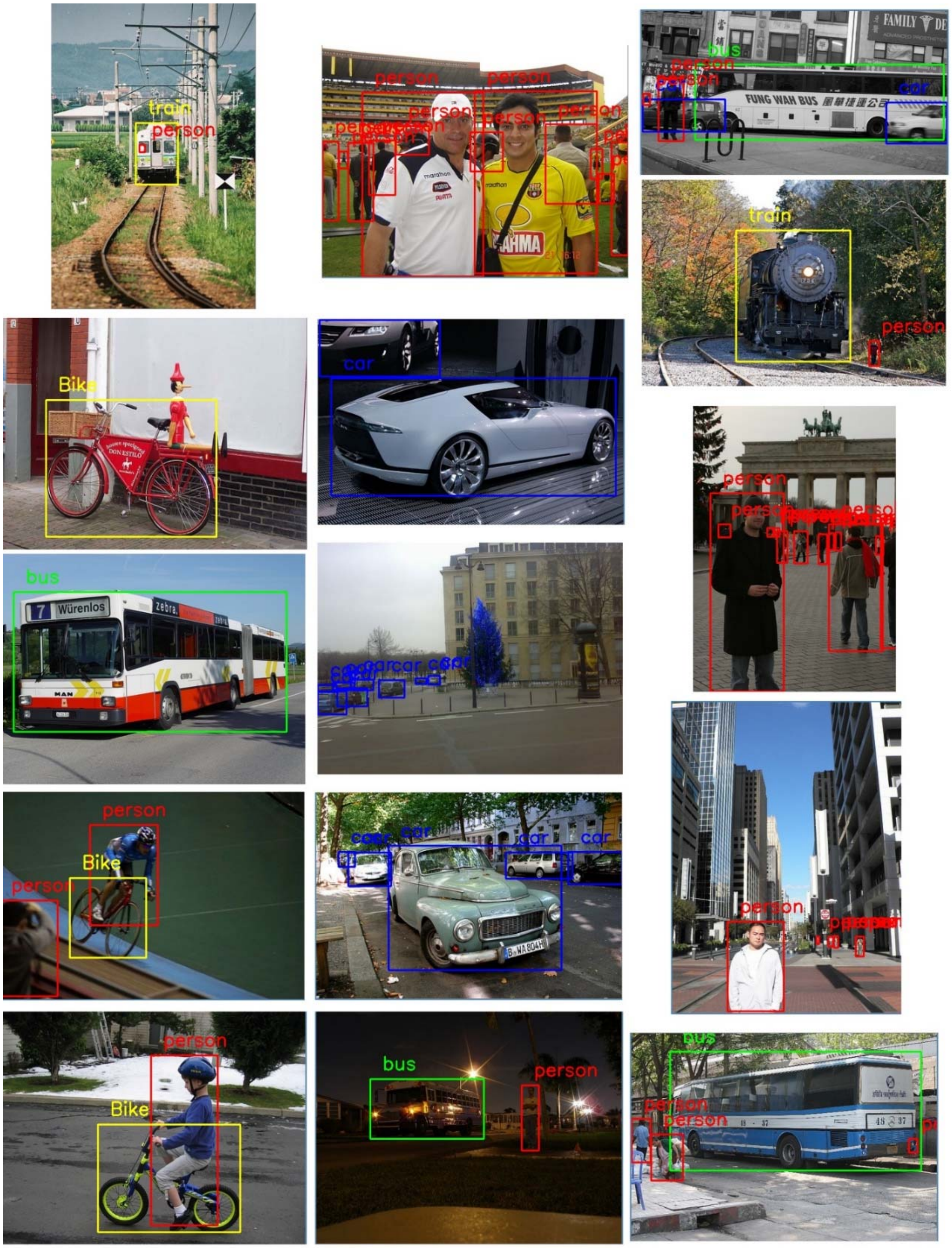

Figure 6: Examples of Detection Results on Pascal VOC Dataset.

improved RPN contains a 1×1 convolution layer for compressing the input channel and a 3×3 dilated convolution layer for increasing the receptive field. Finally, soft-NMS algorithm is adopted to solve the issue of heavy duplicate proposals in traffic scene images. Experimental results on Pascal VOC dataset and KITTI dataset show that the proposed method outperforms Faster R-CNN in detection accuracy. In our future works this paper will investigate more CNN models and enhancements to improve object detection in autonomous driving systems.

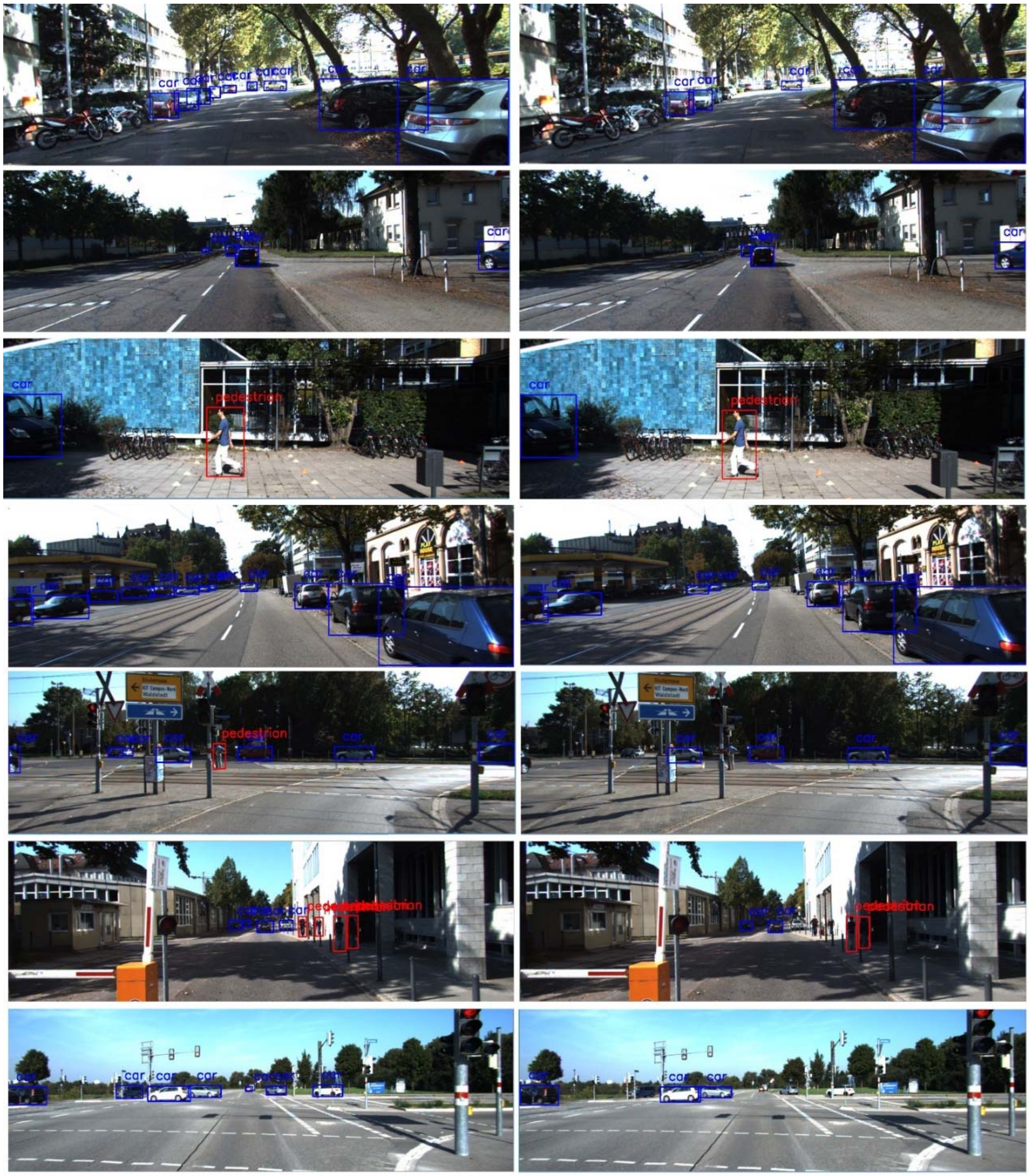

Figure 7: Examples of Detection Results on KITTI Dataset of The Proposed Method (Left Column) and Original Faster R-CNN Framework (Right Column).

REFERENCES:

[1] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." In *Advances in neural information processing systems*, pp. 91-99. 2015.

[2] Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. "Densely connected convolutional networks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700-4708. 2017.

[3] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1-9. 2015.

[4] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[5] Yu, Fisher, and Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions." *arXiv preprint arXiv:1511.07122* (2015).

[6] Wang, Panqu, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. "Understanding convolution for semantic segmentation." In *2018 IEEE winter conference on applications of computer vision (WACV)*, pp. 1451-1460. IEEE, 2018.

[7] Bodla, Navaneeth, Bharat Singh, Rama Chellappa, and Larry S. Davis. "Soft-NMS--improving object detection with one line of code." In *Proceedings of the IEEE international conference on computer vision*, pp. 5561-5569. 2017.

[8] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[9] Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes (voc) challenge." *International journal of computer vision* 88, no. 2 (2010): 303-338.

[10] Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354-3361. IEEE, 2012.

[11] Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. "Feature pyramid networks for object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117-2125. 2017.

[12] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In *European conference on computer vision*, pp. 21-37. Springer, Cham, 2016.

[13] Uijlings, Jasper RR, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. "Selective search for object recognition." *International journal of computer vision* 104, no. 2 (2013): 154-171.

[14] Dai, Jifeng, Yi Li, Kaiming He, and Jian Sun. "R-fcn: Object detection via region-based fully convolutional networks." In *Advances in neural information processing systems*, pp. 379-387. 2016.

[15] Fu, Cheng-Yang, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg. "Dssd: Deconvolutional single shot detector." *arXiv preprint arXiv:1701.06659* (2017).

[16] Zitnick, C. Lawrence, and Piotr Dollár. "Edge boxes: Locating object proposals from edges." In *European conference on computer vision*, pp. 391-405. Springer, Cham, 2014.

[17] Arbeláez, Pablo, Jordi Pont-Tuset, Jonathan T. Barron, Ferran Marques, and Jitendra Malik. "Multiscale combinatorial grouping." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 328-335. 2014.

[18] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886-893. IEEE, 2005.

[19] Lowe, David G. "Distinctive image features from scale-invariant keypoints." *International journal of computer vision* 60, no. 2 (2004): 91-110.

[20] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." *Machine learning* 20, no. 3 (1995): 273-297.

[21] Freund, Yoav, and Robert E. Schapire. "A desicion-theoretic generalization of on-line learning and an application to boosting." In *European conference on computational learning theory*, pp. 23-37. Springer, Berlin, Heidelberg, 1995.

[22] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263-7271. 2017.

[23] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).

[24] Girshick, Ross. "Fast r-cnn." In *Proceedings of the IEEE international conference on computer vision*, pp. 1440-1448. 2015.

[25] Huang, Jonathan, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer et al. "Speed/accuracy trade-offs for modern convolutional object detectors." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7310-7311. 2017.

[26] Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580-587. 2014.

[27] Shrivastava, Abhinav, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. "Beyond skip connections: Top-down modulation for object detection." *arXiv preprint arXiv:1612.06851* (2016).

[28] Yuan, Peng, Yangxin Zhong, and Yang Yuan. "Faster r-cnn with region proposal refinement." *Tech. Rep.* (2017).

[29] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In *Advances in neural information processing systems*, pp. 1097-1105. 2012.