FAST OBJECT DETECTION FRAMEWORK BASED ON MOBILENETV2 ARCHITECTURE AND ENHANCED FEATURE PYRAMID

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

Recently, many object detectors based on deep convolutional neural networks such as Faster R-CNN, SSD, RetinaNet, and so on have been proposed and showed significant improvements over traditional object detectors. However, these deep learning-based object detectors usually focus on detection accuracy. This paper proposes a one-stage deep learning-based object detection framework to improve the inference speed and achieve real-time object detection in outdoor scene images without compromising on accuracy. To improve the inference speed, this paper adopts MobileNet v2 architecture at first to generate the base convolution feature maps. MobileNet v2 achieved comparable performance compared with other state-of-the-art networks while being simpler and faster. To improve the detection accuracy, an enhanced feature pyramid generation module is used to construct rich and multi-scale feature maps from a single resolution input image. Each feature level is a high-level semantic feature map and can be used for detecting objects at a different scale. Finally, a detection network which includes a classification subnet to predict the probability of object presence and a box regression subnet to regress the offset from each anchor box to a nearby ground-truth object is attached to each feature level in the enhanced feature pyramid to locate objects at different scales. In addition, focal loss function is used in the classification branch of the detection network to dedicate the class imbalance problems for the one-stage object detector. Experimental results on public datasets show that the proposed approach achieves nearly as performance as other state-of-the-art approaches, while the inference speed is significant improved.

Keywords: *Deep Learning, Object Detection, Convolutional Neural Networks, MobileNets, Feature Pyramid*

1. INTRODUCTION

There are several fundamental visual recognition problems in the field of computer vision, including image classification, object detection, and semantic segmentation. Image classification aims to recognize semantic categories of objects in each image. Object detection not only recognizes object categories, but also predicts the location of each object in each image by providing a bounding box for each object. Semantic segmentation aims to predict pixel-wise classifiers to assign a specific category label to each pixel, thus providing an even richer understanding of an image. However, semantic segmentation does not distinguish between multiple objects of the same category. Object detection is the basic step towards many computer vision applications such as face recognition, video analysis, vehicle recognition, license plate recognition and so on. Object detection approaches can be divided into two groups: Traditional approaches and deep learning-based approaches. Traditional object detection approaches are usually based on hand-crafted features such as colour, shape, texture, and so on to locate objects in each image. The pipeline of traditional object detection methods is often divided into three steps: proposal generation, feature extraction, and region classification. Most of the successful traditional object detection methods focused on carefully designing feature descriptors to obtain embedding for a region of interest. With the help of good feature representations as well as robust region classifiers, impressive results were achieved on public benchmark datasets. However, traditional object detectors showed many limitations. These limitations included many proposals, redundant proposals, a large number of false positives during classification, difficult to capture representative semantic information in complex contexts, and so on. With the fast development of deep learning, deep learning-based algorithms for object detection outperformed the traditional detection algorithms by a huge margin. In contrast to hand-crafted features used in traditional object detectors, deep convolutional neural networks generate hierarchical feature representations from raw pixels to high level semantic information, which is learned automatically from the training data and shows more discriminative expression capability in complex contexts. Furthermore, benefiting from the powerful learning capacity, a deep convolutional neural network can obtain a better feature representation with a larger dataset, while the learning capacity of traditional visual descriptors are fixed, and cannot improve when more data becomes available. These properties made it possible to design object detection algorithms based on deep convolutional neural networks which could be optimized in an end-to-end manner, with more powerful feature representation capability. Currently, deep learning-based object detection frameworks can be divided into two groups: one-stage frameworks, such as SSD, YOLO and its variants and two-stage frameworks, such as R-CNN and its variants. One-stage frameworks directly make categorical prediction of objects on each location of the feature maps without the cascaded region classification step. Two-stage frameworks first use a proposal generator to generate a sparse set of proposals and extract features from each proposal, followed by region classifiers which predict the category of the proposed region. Two-stage frameworks commonly achieve better detection performance and report state-of-the-art results on public benchmarks, while one-stage detectors are significantly more time-efficient and have greater applicability to real-time object detection. However, current deep learning-based object detection frameworks usually focus on detection accuracy. In the field of computer vision, apart from detection accuracy, inference speed is also a large concern. In this paper, a deep learning-based framework is proposed for fast and efficient object detection. For the purpose of fast object detection on low-end systems, MobileNet v2 architecture is first adopted in this paper to generate the base convolution feature maps. Then, an enhanced feature pyramid generation module is used to create enhanced feature pyramid, where each level of the pyramid can be used for detecting objects at a different scale. Finally, a detection network, which is a fully convolutional network, is attached to each feature level in the enhanced feature pyramid to locate objects. In addition, focal loss function is used in the classification branch of the detection network to dedicate the class imbalance problems for the one-stage object detector.

This paper is organized as follows: an overview of previous methods on object detection 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 Traditional Object Detection Method

Traditional object detection approaches are usually based on hand-crafted features to locate objects in each image. These methods often include three major steps: finding proposals, extracting features, and classification. During finding proposals process, the objective is to search locations in the image which may contain objects. Methods for finding proposals include selective search [10], edge box [11], and multi-scale combinatorial grouping [12]. In order to capture information about multi-scale and different aspect ratios of objects, input images are resized into different scales and multi-scale windows are used to slide through these images. During extracting features process, on each location of the image, a fixed-length feature vector is obtained from the sliding window, to capture discriminative semantic information of the region covered. This feature vector was commonly encoded by low-level visual descriptors such as SIFT [13], Haar [14], HOG [15] or SURF [16], which showed a certain robustness to scale, illumination and rotation variance. During classification stage, the region classifiers are learned to assign categorical labels to the covered regions. Classification methods include support vector machine (SVM) [17], adaboost [18], and so on. Although traditional methods showed a good performance on many public benchmark datasets, they still had many limitations in difficult conditions. First, a large number of proposals are generated during proposal generation, and many of them are redundant. This resulted in a large number of false positives during classification. Second, feature descriptors are hand-crafted based on low level visual cues, which made it difficult to capture representative semantic information in complex conditions. Finally, each step of the detection pipeline is designed and optimized separately, and thus could not obtain a global optimal solution for the whole system.

2.2 Deep Learning-based Object Detection Method

After the success of applying deep convolutional neural networks (CNNs) for image classification, object detection also achieved remarkable progress based on deep CNNs. One-stage and two-stage network architectures are predominantly used in general object detection. The input images to the one-stage architecture are sliced into several grid cells. The classifier outputs a vector that encodes the information of each grid cell. OverFeat [19] was one of the first modern one-stage object detector framework based on deep CNNs that achieved great success on general object detection. Recently, SSD [20] and YOLO [21] have renewed interest in one-stage methods. These detectors have been tuned for speed but their accuracy trails that of two-stage methods. SSD [20] uses default boxes of different scales to multiple layers within a ConvNet and enforces each layer to focus on predicting objects of certain scale. To improve detection accuracy on multi-scale layer, MS-CNN [22] applies deconvolution on multiple layers of a ConvNet to increase feature map resolution before using the layers to learn region proposals and pool features. The two-stage architecture first finds the ROIs and then performs detection in every ROI. The first two-stage CNN-based object detection framework was R-CNN [23], in which the selective search algorithm is used to find possible proposals. Convolution is used to extract the features, following which a representative vector was sent to the final SVM classifier. However, this method is very time-consuming, as every proposal is processed by the entire CNN. To improve upon the speed performance, Fast R-CNN [24] was developed. This approach is based on ROI-Pooling, in which the feature maps are shared after convolution. Therefore, only one convolution is processed in the entire detection model. The problem with Fast R-CNN is that the proposals are generated by a traditional time-consuming selective search. Faster R-CNN [4] was proposed to further improve upon Fast R-CNN. In Faster R-CNN, an RPN is used to replace the selective search. The proposal task is converted into a binary problem, in which the network predicts if an object is present in a box. Therefore, in this approach, a small subnetwork, the RPN, is used to solve the problem. Subsequently, the generated proposals are processed using Fast R-CNN. The important point is that the inputs to the RPN comprise the feature maps of convolution, so there is slightly more loading involved when using the RPN. In terms of the inference speed, there is a great improvement from R-CNN to Faster R-CNN. Fast R-CNN is 146 times faster than R-CNN, and Faster R-CNN is 1460 times faster than R-CNN.

Recently, several studies have proposed techniques to increase the accuracy of Faster R-CNN. Instead of using VGG-16 as a backbone network for Faster R-CNN, adoption of different backbone networks, such as ResNet and Inception ResNet, has been proposed. He et al. [25] 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. [9] 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. Dai et al. [5] proposed the use of a Region-based Fully Convolution Network (R-FCN) 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. Huang et al. [26] 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 previous methods with a slightly higher frame rate. Yauan et al. [28] proposed two refinement methods, iterative and LSTM refinement, for the Faster R-CNN model and improved the accuracy. Most of the above-mentioned works focus on increasing accuracy instead of improving the frame rate. Shih et al. [29] proposed a method to both accelerate the frame rate and improve the detection accuracy. DSSD [30] proposed an approach for introducing additional context into state-of-the-art general object detection.

3. PROPOSED APPROACH

Figure 1 illustrates the overall architecture of the proposed approach. As shown in Figure 1, for the purpose of fast object detection on low-end systems, MobileNet v2 architecture is first adopted to generate the base convolution feature maps. An enhanced feature pyramid generation module is then designed to create enhanced feature pyramid, where each level of the pyramid can be used for detecting objects at a different scale. Finally, a detection network, which is a fully convolutional network, is attached to each feature level in the enhanced feature pyramid to locate objects. In addition, focal loss

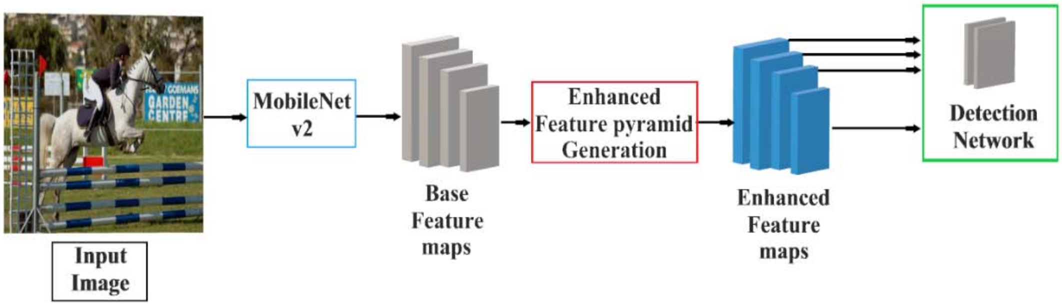

Figure 1: The Overall Architecture of The Proposed Method.

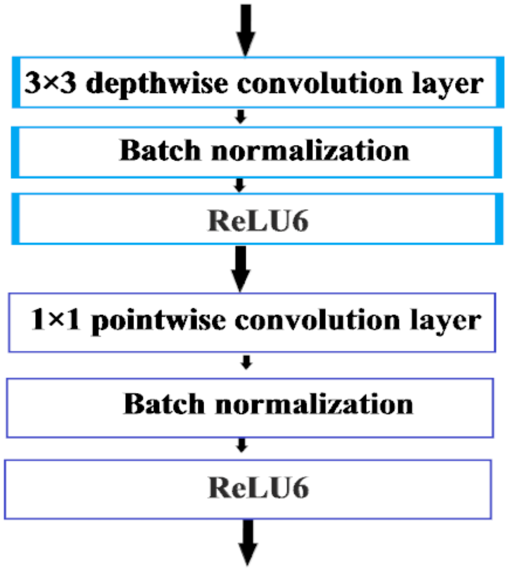
3×3 depthwise convolution layer
Batch normalization
ReLU6
1×1 pointwise convolution layer
Batch normalization
ReLU6

Figure 2: The Architecture of a Depthwise Separable Convolution Block.

function is used in the classification branch of the detection network to dedicate the class imbalance problems for the one-stage object detectors. Details of each module will be explained in the following sections.

3.1 Initial Feature Maps Generation

MobileNet is a neural network architecture introduced by Google that runs very efficiently on mobile devices or any devices with low computational power. MobileNet was created as a model that delivered high accuracy while keeping the parameters and mathematical operations as low as possible. The MobileNet v1 [31] architecture replaces regular convolutional layers, which are essential to computer vision tasks but are quite expensive to compute, by depthwise separable convolutions. Depthwise separable convolution includes a 3×3 depthwise convolution layer that filters the input feature and a 1×1 convolution layer called pointwise convolution layer that combines these filtered values to create new features. Figure 2 shows the architecture of a depthwise separable convolution block. Each of the depthwise and pointwise convolution layers is followed by batch normalization and ReLU6 layer as follow:

$$y = \min(\max(0, x), 6) \tag{1}$$

Together, the depthwise and pointwise convolution layers do approximately the same thing as regular convolution layers but are much faster. Supposing that the size of input image is 224×224×3, the full architecture of MobileNet v1 is shown in Table 1. As

Table 1: The Full Architecture of MobileNet v1.

Layer	Type	Output Size
0	3×3 regular convolution layer	112×112×32
1	depthwise separable convolution block	112×112×64
2	depthwise separable convolution block	56×56×64
3	depthwise separable convolution block	56×56×128
4	depthwise separable convolution block	28×28×128
5	depthwise separable convolution block	28×28×256
6	depthwise separable convolution block	14×14×256
7	depthwise separable convolution block	14×14×512
8	depthwise separable convolution block	14×14×512
9	depthwise separable convolution block	14×14×512
10	depthwise separable convolution block	14×14×512
11	depthwise separable convolution block	14×14×512
12	depthwise separable convolution block	7×7×512
13	depthwise separable convolution block	7×7×1024

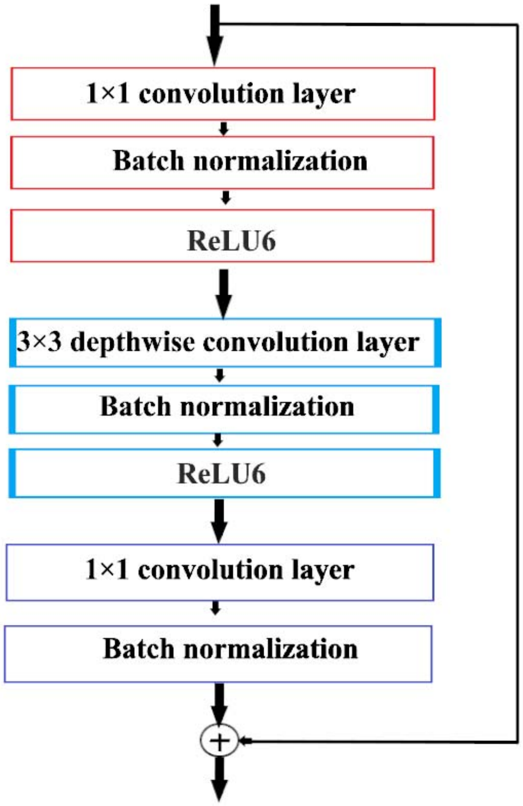

Figure 3: The Architecture of The Depthwise Separable Convolution Block Used in MobileNet v2 Architecture.

shown in Table 1, MobileNet v1 consists of a regular 3×3 convolution layer as the first layer, followed by 13 depthwise separable convolution blocks. There are no pooling layers in between these depthwise separable blocks. Instead, some of the depthwise

Table 2: The Full Architecture of MobileNet v2.

Layer	Type	Output Size
0	3×3 regular convolution layer	112×112×32
1	new depthwise separable convolution block	112×112×16
2-3	new depthwise separable convolution block	56×56×24
4-6	new depthwise separable convolution block	28×28×32
7-10	new depthwise separable convolution block	14×14×64
11-13	new depthwise separable convolution block	14×14×96
14-16	new depthwise separable convolution block	7×7×160
17	new depthwise separable convolution block	7×7×320
18	1×1 regular convolution layer	7×7×1280

convolution layers have a stride of 2 to reduce the spatial dimensions of the input feature map.

The MobileNet v2 [32] is introduced as a refinement of the MobileNet v1 that makes it even more efficient and powerful. In the MobileNet v2 architecture, the depthwise separable convolution block has been re-designed as shown in Figure 3. There are three convolutional layers in the new depthwise separable convolution block. The first layer is a 1×1 convolution layer which is used to expand the number of channels in the input feature map before it goes into the depthwise convolution layer. The middle layer is a 3×3 depthwise convolution layer that filters the input feature map as in the MobileNet V1. The final layer is a 1×1 convolution layer. However, this final convolution layer is used to project data with a high number of channels into a tensor with a much lower number of channels, thus making the number of channels of the input feature map smaller. This final layer is also called a bottleneck layer because it reduces the amount of data that flows through the network.

Furthermore, the residual connection as in ResNet [25] is adopted in the MobileNet v2 architecture to help with the flow of gradients through the network. Each layer in the new depthwise separable convolution is followed by batch normalization and ReLU6 as activation function (except the last bottleneck layer since using a non-linearity after this layer destroyed useful information). The full MobileNet v2 architecture is shown in Table 2. As shown in Table 2, MobileNet v2 consists of 17 of the new depthwise separable convolution blocks followed by a regular 1×1 convolution layer. The first layer is a regular 3×3 convolution layer with 32 channels.

For the purpose of fast object detection on low-end systems, this paper adopts MobileNet v2 architecture to generate initial feature maps. The feature maps generated after layer 18, layer 13, layer 6, and layer 3 are adopted to generated enhanced feature maps by feature pyramid generation module as explained in the next section.

3.2 Enhanced Feature Pyramid Generation

This paper adopts the Feature Pyramid Network (FPN) from [9] to build the enhanced feature pyramid generation module which generates enhanced feature maps with different resolutions. FPN augments a standard convolutional network with a top-down pathway and lateral connections so the network efficiently constructs a rich, multi-scale feature pyramid from a single resolution input image. Each level of the pyramid can be used for detecting objects at a different scale. Following [20], this paper builds FPN on top of the MobileNet v2 architecture as shown in Figure 4. Let {C5, C4, C3, C2} denotes the output feature map of layer 18, layer 13, layer 6, and layer 3 of the MobileNet v2. This paper first applies a 1×1 convolution layer to reduce C5 channel depth to 256-d to create M5. This feature map then becomes the first feature map layer used for object detection in the detection network. Following the top-down path, this paper upsamples the previous layer by a factor of 2 using nearest neighbors upsampling algorithm. At the same branch, a 1×1 convolution layer is applied to each corresponding feature maps in the base network. Then, the corresponding feature map is merged with the upsampled feature map by element-wise addition. A 3×3 convolution layer is applied to all merged feature maps to reduce the aliasing effect when merged with the upsampled layer. This process is iterated until the finest resolution map is generated. The final set of feature maps is called {P5, P4, P3, P2}, corresponding to {C5, C4, C3, C2} that are respectively of the same spatial sizes. Because the above process shares the same classifier and box regressor of every output feature maps, all

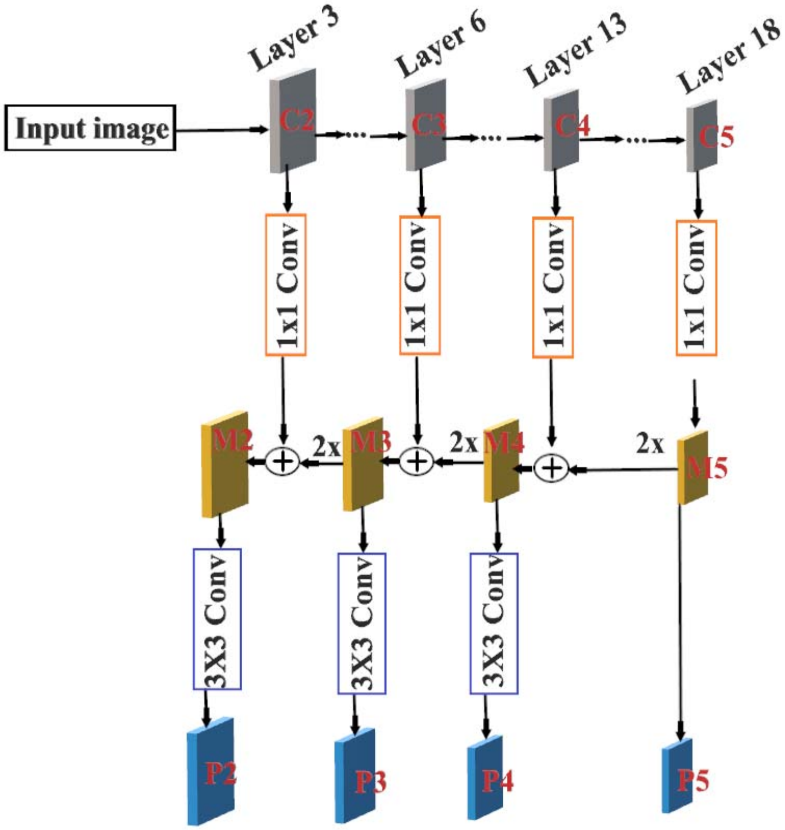

Figure 4: The Architecture of The Enhanced Feature Pyramid Generation Module.

pyramid feature maps P5, P4, P3 and P2 have 256-d output channels.

For anchor boxes, because the detection network slides over all locations in all pyramid levels of the enhanced feature pyramid, it is not necessary to have multi-scale anchor boxes on a specific level. Thus, this paper assigns anchor boxes of a single scale to each level of the enhanced feature pyramid as in [9]. More specific, this paper defines the anchor boxes to have areas of {32×32, 64×64, 128×128, 256×256} pixels on {P2, P3, P4, P5} respectively. For aspect ratio, this paper also uses anchor boxes of multiple aspect ratios {1:2, 1:1, 2:1} at each level of the enhanced feature pyramid. Thus, there are 15 anchor boxes at each location over the enhanced feature pyramid. For training samples, this paper assigns training labels to the anchor boxes based on their Intersection-over-Union (IoU) ratios with ground-truth bounding boxes as in [4]. More specific, an anchor box is assigned a positive label if it has the highest IoU for a given ground-truth box or an IoU over 0.7 with any ground-truth box, and a negative label if it has IoU lower than 0.3 for all ground-truth boxes.

3.3 Detection Network

The detection network includes a classification subnet and a box regression subnet as shown in Figure 1. The classification subnet predicts the probability of object presence at each location of each level of the enhanced feature pyramid for each of anchor box and object class. The proposed classification subnet is a small fully convolutional network attached to each feature level in the enhanced feature pyramid. The parameters of the classification subnet are shared across all pyramid levels. Figure 5 (a) shows the architecture of the proposed classification subnet. For each input feature map with C channels, four 3×3×C convolution layers followed by ReLU activation are first applied. Then, a 3×3 convolution layer with NA filters, where N denotes the number of object classes and A denotes the number of anchor boxes, is used. Finally, sigmoid activations are attached to output

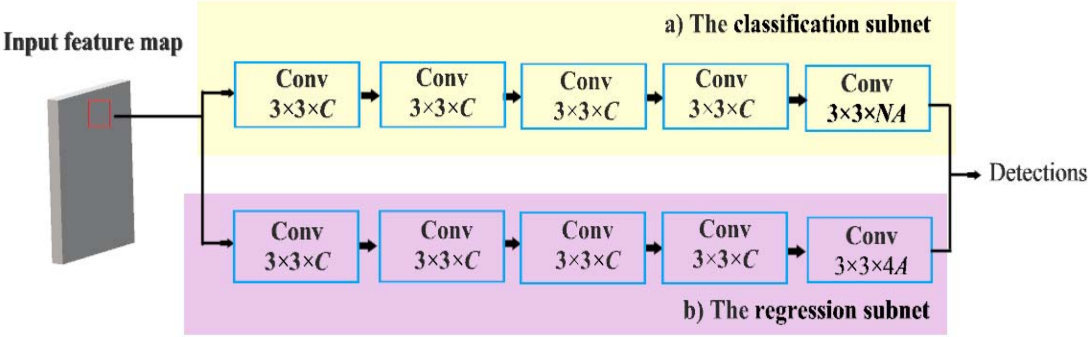

Figure 5: The Architecture of The Detection Network. a) The Classification Subnet. b) The Box Regression Subnet.

NA binary predictions per spatial location. The regression subnet is used to regress the offset from each anchor box to a nearby ground-truth object. Figure 5 (b) shows the architecture of the proposed regression subnet. As shown, the architecture of the box regression subnet is similar to the architecture of the proposed classification subnet. However, the regression subnet outputs *4A* linear values per spatial location. For each of the *A* anchor boxes per spatial location, these four outputs predict the relative offset between the anchor box and the ground-truth box. The object classification subnet and the box regression subnet, though sharing a common structure, use separate parameters.

3.4 Loss Function

The Cross Entropy (CE) loss function [4] is the common loss function used for object classification. The CE loss can reduce the imbalance between positive and negative samples. However, the CE loss is not good enough to train classifier for distinguishing between easy and hard samples. In general object detection, because of complex backgrounds, the problem of balance between easy and hard samples becomes more significant. Focal loss function (FC) is original introduced by Lin et al. [33] to dedicate the class imbalance problems for the one-stage object detectors. Inspired by the improvements of the FC loss, this paper proposes to use the focal loss function instead of the conventional CE loss. For better understanding, let's have a quick review on the CE loss function.

The traditional CE loss for classification is formally defined as follow:

$$L_{CE}(p, y) = -\log(p_t) \tag{2}$$

$$p_t = \begin{cases} p & if\ y = 1 \\ 1 - p & otherwise \end{cases} \tag{3}$$

where p represents the predicted probability of given candidate having label +1; $y \in \{-1, +1\}$ represents the ground-truth label.

By adding a modulating factor $(1 - p_t)^\gamma$ to the CE loss, where $\gamma \geq 0$ represents the tunable focusing parameter. The loss function becomes the FL loss. Thus, the FL loss function is defined as follow:

$$L_{FL}(p_t) = -(1 - p_t)^\gamma \log p_t \tag{4}$$

With the FL loss function replaces the traditional CE loss function, the contribution of the easy examples is reduced while the ones from hard examples are enhanced during the training process.

4. EXPERIMENTAL RESULTS

In this section, this paper provides details of the experimental results of the proposed method, and then compares the results of the proposed approach with other approaches on general object detection.

4.1 Experimental Environment and Datasets

The proposed method is implemented in TensorFlow with a Python interface. The CPU used for training and testing process is Intel Core i7-8700, the main memory is RAM with 16 GB DDR4, and the GPU is NVIDIA GeForce GTX 1080.

In this paper, two popular datasets are adopted to evaluate the proposed approach, including PASCAL VOC 2007 dataset [1] and KITTI dataset [2].

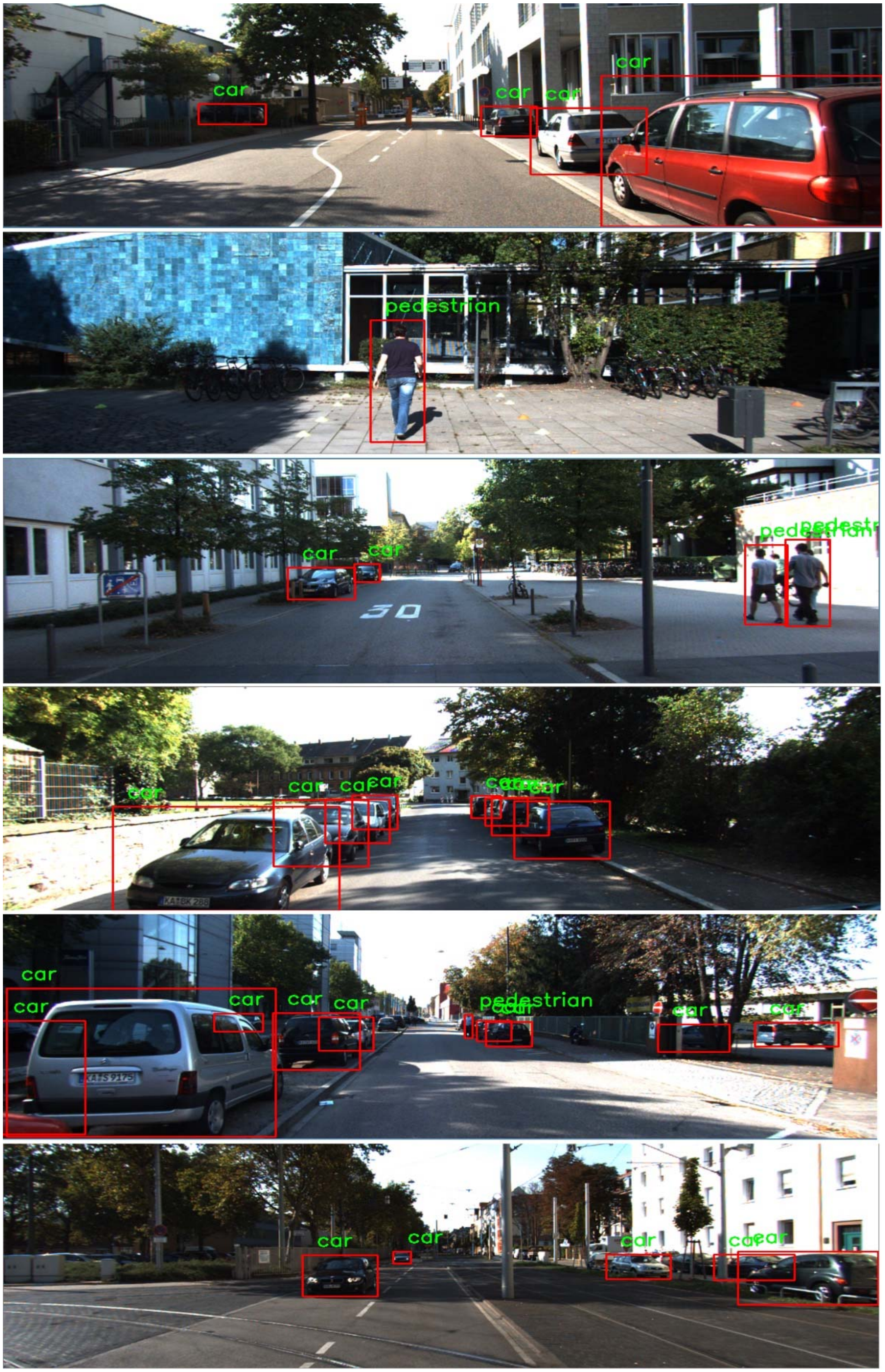

Figure 6: Detection Examples on The KITTI Dataset.

PASCAL VOC 2007 dataset includes 5011 images in the training set and 4952 images in the testing set. This dataset has 20 classes, such as vehicles, animals, indoor objects and person. Following the instructions in [1], this paper uses the mAP@0.5 metric to evaluate accuracy. This paper does not change the training set and testing set for cross validation. However, for each of the mAP values in the experiments, this paper runs the simulation approximately eight times and use the average mAP as the results. In all experiments, this paper adopts the pre-trained networks on ImageNet [3]. The proposed model is trained for 90k iterations with an initial learning rate of 0.01, which is then divided by 10 at 60k and again at 80k iterations. Weight decay of 0.0001 and momentum of 0.9 are used. The training loss is the sum the focal loss and the standard smooth L1 loss used for box regression [4].

KITTI dataset is a large driving environment dataset. In this paper, the 2D object detection dataset is used for evaluation. In the proposed experiment, this paper redefines the object classes into two classes: car and pedestrian. Car includes Van, Truck, Car, Tram and pedestrian includes Pedestrian, Cyclist. KITTI dataset provides 7481 images for training and 7518 for testing.

4.2 Object Detection Results on PASCAL VOC 2007 Dataset

In this section, this paper compares the detection results of the proposed method with other methods on PASCAL VOC 2007 Dataset, including Faster R-CNN [4], R-FCN [5], CoupleNet [6], ERPN [7], and method proposed by He et al. [8]. Faster R-CNN [4] introduced a Region Proposal Network that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. R-FCN [5] proposed a region-based detector which is fully convolutional with almost all computation shared on the entire image. CoupleNet [6] proposed a novel fully convolutional network to couple the global structure with local parts for object detection. ERPN [7] proposed an Enhanced Region Proposal Network which includes deconvolutional feature pyramid network, novel anchor box setting, a particle swarm optimization, and multi-task loss function for object detection. He et al. [8] introduced a new channel pruning method to accelerate very deep convolutional neural networks.

Table 3 shows the comparison of detection results of the proposed method and other methods on PASCAL VOC 2007 dataset. As shown in Table 3, the proposed approach outperforms Faster R-CNN, ERPN, and method proposed by He et al. [8] on PASCAL VOC 2007 dataset. More specific, the mAP of the proposed approach is improved by 10.2%, 0.3%, 12% compared with Faster R-CNN, ERPN, and method proposed by He et al. [8] respectively. For the inference time, the proposed approach achieves the best inference speed with only 0.03 second to process an image. Among the reference methods, CoupleNet achieves the best detection result. However, CoupleNet takes up to 0.13 second to process an image. Thus, CoupleNet is slower than the proposed method. It should be noticed that CoupleNet is a two-stage detector while the proposed detector is a one-stage detector. The results in Table 3 show that the proposed detector provides an optimal trade-off between detection accuracy and speed by providing a detection accuracy of 78.9 mAP while running at 33 FPS.

4.3 Object Detection Results on KITTI Dataset

To further evaluate the performance of the proposed method, this paper conducts experiments on the KITTI dataset. The detection results are compared with FPN [9] with RPN baseline. FPN exploited the inherent multi-scale, pyramidal hierarchy of deep convolutional networks to construct feature pyramids with marginal extra cost. Table 4 shows the detection results on the KITTI dataset of the proposed method and FPN with RPN baseline. Both methods in Table 4 are tested and trained on KITTI with a 0.5 IoU threshold. As shown in Table 4, the mAP of the proposed method is 2.5 points higher than FPN. Especially, the AP of pedestrian detection, which contains more small-size objects, is increased by 4.1 points. The results demonstrate that the proposed method achieves better performance for small-size objects in difficult conditions. Figure 4 shows several detection examples on the KITTI dataset. The detection results in Figure 4 show that the proposed method can exactly locate small and occluded objects.

5. CONCLUSIONS

Fast and accurate object detection is one of the most critical problems in the field of computer vision. Recently, convolutional neural networks achieved huge successes on visual object detection over traditional object detectors, which use hand-crafted features. However, due to the challenging in difficult conditions such as large object scale variation, object occlusion and bad light conditions, popular deep CNN-based detectors such as Faster-RCNN, SSD, YOLO, and so on do not produce good detection performance over large benchmark

Table 3: Comparison of detection results on PASCAL VOC 2007 Dataset.

Method	Base Network	mAP (%)	Inference Time (s)
Faster R-CNN [4]	VGG-16	68.7	0.2
R-FCN [5]	ResNet-101	79.5	0.08
CoupleNet [6]	ResNet-101	82.7	0.13
ERPN [7]	VGG-16	78.6	0.17
He et al. [8]	VGG-16	66.9	0.05
Proposed Method	MobileNet v2	78.9	0.03

Table 4: Detection Results on The KITTI Dataset.

Method	AP (%)		mAP (%)
	Car	Pedestrian	
FPN with RPN baseline	90.3	78.3	84.3
Proposed Approach	91.1	82.4	86.8

datasets. This paper proposes a deep learning-based framework for fast and efficient object detection in difficult conditions based on MobileNet v2 architecture and enhanced feature pyramid. The proposed framework is a one-stage architecture to improve the inference speed and achieve real-time object detection in outdoor scene images without compromising on accuracy. To improve the inference speed, MobileNet v2 architecture is used to generate the base convolution features. To improve the detection accuracy, an enhanced feature pyramid generation module is used to construct rich and multi-scale feature maps from a single resolution input image. Finally, a detection network which includes a classification subnet to predicts the probability of object presence and a box regression subnet to regress the offset from each anchor box to a nearby ground-truth object is attached to each feature level in the enhanced feature pyramid to locate objects at different scales. In addition, focal loss function is used in the classification branch of the detection network to dedicate the class imbalance problems for the one-stage object detectors. The proposed framework is evaluated by extensive experiments over PASCAL VOC 2007 dataset and KITTI dataset. The effectiveness of the proposed method was verified by experimental results with comparable detection performance over benchmark datasets. In the future works, this paper will investigate more enhancements to improve object detection for fast and efficient intelligent transport systems.

REFERENCES:

[1] 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, Sep. 2009.

[2] Geiger, A.; Lenz, P.; Urtasun, R., “Are we ready for autonomous driving? The kitti vision benchmark suite,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 16–21 June 2012.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in Proc. Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

[4] 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.

[5] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[6] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu, “CoupleNet: Coupling global structure with local parts for object detection,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Oct. 2017, pp. 4126–4134.

[7] Y. P. Chen, Y. Li, and G. Wang, “An enhanced region proposal network for object detection using deep learning method,” *PLoS ONE*, vol. 13, no. 9, Sep. 2018, Art. no. e0203897.

[8] Y. He, X. Zhang, and J. Sun, “Channel pruning for accelerating very deep neural networks,” in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, vol. 2, no. 6, pp. 1389–1397.

[9] 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.

[10] 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, Sep. 2013.

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

[12] P. Arbelaez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, “Multiscale combinatorial grouping,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 328–335.

[13] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[14] R. Lienhart, J. Maydt, “An extended set of haar-like features for rapid object detection,” in *International Conference on Image Processing*, 2002.

[15] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, vol. 1, no. 1, pp. 886–893.

[16] H. Bay, T. Tuytelaars, L. Van Gool, “Surf: Speeded up robust features,” in *ECCV*, 2006.

[17] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[18] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[19] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” In *ICLR*, 2014.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, “SSD: Single shot multibox detector,” In *ECCV*, 2016.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” In *CVPR*, 2016.

[22] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, “A unified multi-scale deep convolutional neural network for fast object detection,” In *ECCV*, 2016.

[23] 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.

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

[25] 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.

[26] J. Huang et al., “Speed/accuracy trade-offs for modern convolutional object detectors,” in *Proc. CVPR*, vol. 4, Jul. 2017, pp. 3296–3297.

[27] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta, “Beyond skip connections: Top-down modulation for object detection,” Dec. 2016, arXiv:1612.06851. [Online]. Available: https://arxiv.org/abs/1612.06851

[28] P. Yuan, Y. Zhong, and Y. Yuan, “Faster R-CNN with region proposal refinement,” *Comput. Sci. Dept., Stanford Univ., Tech. Rep.*, 2017.

[29] K.-H. Shih, C.-T. Chiu, and Y.-Y. Pu, “Real-time object detection via pruning and a concatenated multi-feature assisted region proposal network,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2019, pp. 1398–1402.

[30] Fu, Cheng-Yang, et al. "Dssd: Deconvolutional single shot detector." arXiv preprint arXiv:1701.06659 (2017).

[31] Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.04861 (2017).

[32] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510-4520. 2018.

[33] Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," In *Proceedings of the IEEE international conference on computer vision*, pp. 2980-2988. 2017.