Comprehensive Feature Enhancement Module for Single-Shot Object Detector

Qijie Zhao, Yongtao Wang*, Tao Sheng, and Zhi Tang

Institute of Computer Science and Technology, Peking University, Beijing, China
{zhaoqijie, wyt, shengtao, tangzhi}@pku.edu.cn

Abstract. Recent one-stage CNN based detectors attract lots of research interests due to the high detection efficiency. However, their detection accuracy usually is not good enough. The major reason is that with only one regression step, one-stage detectors like SSD build features which are not representative enough for both classification and localization. In this paper, we propose a novel module, Comprehensive Feature Enhancement(CFE) module, for largely enhancing the features of one-stage detectors. The effective yet lightweight module could improve the detection accuracy with only increasing little inference time. Moreover, we propose two new one-stage detectors by assembling CFEs into the original SSD: CFE-SSDv1 and CFE-SSDv2. The CFE-SSDv1 is of simple structure with high efficiency while CFE-SSDv2 is more accurate and improves dramatically on detecting small objects especially. We evaluate the proposed CFE-SSDv1 and CFE-SSDv2 on two benchmarks for general object detection: *PASCAL VOC07* and *MS COCO*. Experimental results show that CFE-SSDv2 outperforms state-of-the-art one-stage methods such as DSSD and RefineDet on these two benchmarks. Moreover, additional ablation study demonstrates the effectiveness of the proposed CFE module. We further test the proposed CFE-SSDv2 on *UA-DETRAC* dataset for vehicle detection and *BDD* dataset for road object detection, and both get accurate detection results compared with other state-of-the-art methods.

1 Introduction

Object detection is one of the most fundamental problems in the field of computer vision, and recent deep neural network (DNN) based methods achieve state-of-the-art results for this problem. The state-of-the-art methods for general object detection can be briefly categorized into one-stage methods (e. g.,YOLO [24], SSD [22], Retinanet [19], DSSD [5], RefineDet [36]), and two-stage methods (*e.g.*, Fast/Faster R-CNN [27], FPN [18], Mask R-CNN [9]). Generally speaking, two-stage methods usually have better detection performance while one-stage methods are more efficient.

For many real world applications, *e.g.*, video surveillance and autonomous driving, an adequate object detector should be effective and efficient enough, and

* Corresponding Author

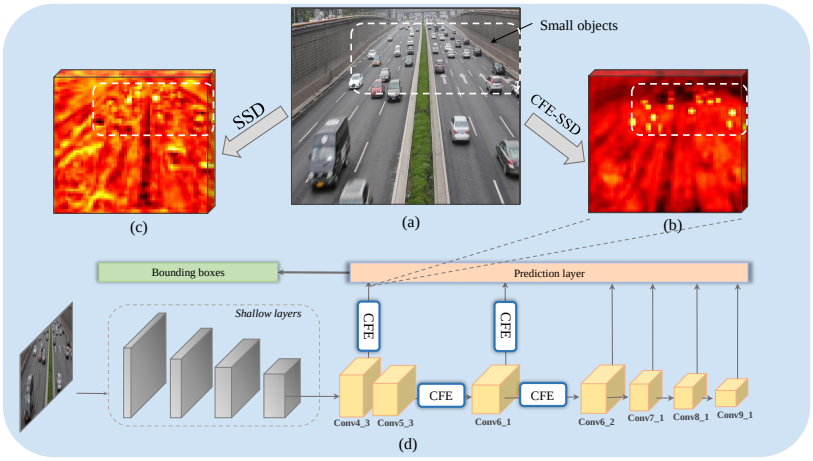

Fig. 1. CFE-SSDv1 effectively enhance the corresponding features for small objects. (a) An image for testing. (b),(c) The feature map for detecting small objects respectively of CFE-SSDv1 and the original SSD. (d) The structure of CFE-SSDv1.

has strong ability to detect small objects. For example, regarding the application of autonomous driving, the object detector should be very effective to sense the surrounding scenes and also should be very efficient to avoid obstacles in time. Moreover, the ability of detecting small objects is very important in autonomous driving scenes, due to that large portion of small objects appears in these scenes, such as traffic lights, traffic signs and faraway objects. However, state-of-the-art detectors cannot fulfill all the above requirements. Hence, in this work, we make a first attempt to propose a detector can fit all these requirements.

To achieve this goal, we improve the most widely used one-stage detector SSD by enhancing the CNN features for predicting candidate detections, based on our observations: (1) the shallower layer features used to detect small objects doesn't contain rich high-level semantic information thus not discriminative enough for the classification task, and (2) the deeper layer features used to detect large objects are less position-sensitive thus not accurate enough for the bounding box regression task. To be more specific, we propose a novel Comprehensive Feature Enhancement(CFE) module and assemble four CFE modules into the network architecture of SSD for enhancing the CNN features. As illustrated in Figure 1, CNN features corresponding to small objects are effectively enhanced by the proposed CFE modules. Moreover, experimental results show that CFE module can also promote the detection performance of other one-stage detectors like DSSD [5] and RefineDet [36].

The main contributions of this work are summarized as follows.

– We introduce Comprehensive Feature Enhancement (CFE) module, an effective and flexible module for learning better CNN features to improve the detection accuracy of the single shot object detectors.

– Based on the proposed CFE module, we further propose two one-stage detectors, CFE-SSDv1 and CFE-SSDv2, which are efficient as SSD while have much better detection accuracy than SSD, especially for small objects.
– The proposed CFE-SSDv2 achieves good results on multiple benchmark datasets: outperforms the state-of-the-art one-stage detectors DSSD and RefineDet on VOC07 and MS-COCO datasets for general object detection, gains the best result on UA-DETRAC [33] dataset for vehicle detection, and the second best result on BDD [35] dataset for road object detection.

2 Related Work

Due to DNN based methods become dominate now, we only review them in this section, which can be briefly divided into two groups: two-stage detectors and one-stage detectors.

Two-stage Detectors. The two-stage detectors consist of a proposal generation stage (*e.g.*, Selective Search [32] for Fast R-CNN [7] and RPN for Faster R-CNN [27]) and a stage for object classification and bounding box regression. The two-stage detectors (*e.g.*, R-CNN [8], Fast R-CNN [7], Faster R-CNN [27], R-FCN [3], HyperNet [15], FPN [18], Mask R-CNN [9], PANet [21]) keep achieving state-of-the-art performance on several challenging datasets such as PASCAL VOC 2007, 2012 [4] and MS COCO [20]). However, their efficiency is not high enough, which is the main drawback of them.

One-Stage Detectors. The one-stage detectors directly perform object classification and bounding box regression over a regular, dense sampling of object locations, scales, and aspect ratios, that is, skip the proposal generation stage. One-stage detectors usually have notably higher efficiency than two-stage detectors. YOLO [24] and SSD [22] are two representative one-stage detectors. YOLO adopts a relatively simple architecture thus very efficient, but cannot deal with dense objects or objects with large scale variants. As for SSD, it could detect objects with different sizes from multi-scale feature maps. Moreover, SSD uses anchor strategy to detect dense objects. Therefore, it achieves a pretty detection performance. In addition, SSD with input size of 512*512 can achieve the speed of about 23 FPS on the graphics processing unit (GPU) such as Titan XP. Due to the above advantages, SSD becomes a very practical object detector in industry, which has been widely used for many tasks. However, its detection performance is still not good enough, especially for small objects. For example, on the test-dev set of MSCOCO [20], the average precision(AP) of small objects of SSD is only 10.9%, and the average recall(AR) is only 16.5%. The major reason is that it only uses shallow feature maps to detect small objects, which doesn't contain rich high-level semantic information thus not discriminative enough for classification. Recently proposed one-stage detector RetinaNet [27] show comparable detection performance with the state-of-the-art two-stage detectors. However, its efficiency is not good when the best detection performance is achieved. More recently proposed RefineDet [36]performs best among the existing one-stage detectors, in both terms of detection accuracy and speed. RefineDet uses an Encode-Decode

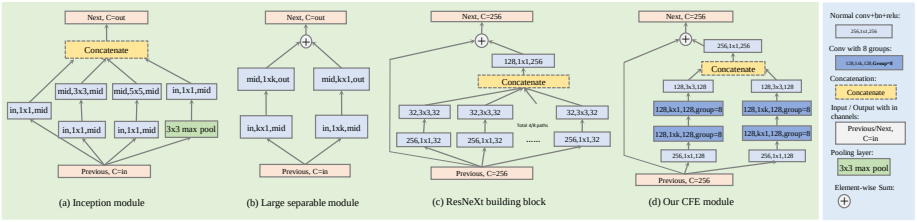

Fig. 2. Example structures for learning representative features: (a) the Inception module [30], (b) the large separable module [16], (c) the ResNeXt module [34] and (d) our CFE module. A Conv represents a combination of Conv+BatchNorm+ReLU layers.

[5] structure to deepen the network and upsample feature maps so that large-scale feature maps can also learn deeper semantic information. On the other hand, RefineDet uses the idea of cascade regression like Faster-RCNN [27], applying the Encode part to firstly regress coarse positions of the targets, and then use the Decode part to regress out a more accurate position on the basis of the previous step. On MSCOCO test-dev, it gets the average precision of 16.3% and an average recall of 29.3% for small objects. Also, RefineDet with VGG backbone could perform with high efficiency. Although the result achieved by RefineDet is much better, there is still much room for performance improvement.

CNN Structures. In addition, we also review some developments of CNN structure which are also introduced to our CFE module. First, Convolution(Conv) layers have powerful feature learning capabilities, and stacking multiple 3x3 Conv layers can capture advanced semantics [29]. Inspired by NIN [17], Inception module(in Figure 2.a) proposes the split-transform-merge method, decomposing a Conv layer into multiple joint 1×1 and k×k conv layers combinations. Moreover, Xception [2] and MobileNet [12] take use of connecting depthwise separable convolution and pointwise convolution for weight lighting.

Second, it is necessary to expand the scope of the proximity relationship and learn the high-level semantics from large receptive field. Inside an upgraded version of Inception, like Inception V3 [31], Conv layers with kernel size=7 are applied for expanding the receptive field. Specifically, after factorization, continuously connecting 1x7 and 7x1 Conv layers can decrease parameter magnitude, while still keeping the area of its receptive field. And shown in Figure 2.b, the large kernel convolution is proposed in [23] to improve semantic segmentation, the large separable module [16] is also implemented after feature extraction in object detection system.

Third, decreasing redundancy is also of great significance. ResNet [11] introduces the Bottleneck structure to achieve this target. Additionally, as shown in Figure 2.c, ResNeXt [34] proposes to group convolution kernels. Specifically, the aggregated transformations of ResNeXt module replace the Conv layers of original ResNet module with blocks of the same topology stacked in parallel. This scheme improves the model without increasing the parameter magnitude.

3 Comprehensive Feature Enhancement(CFE) module

Our target is to design a module for learning better features, which should have advantages in terms of receptive field and learning capacity, also have less weight redundance and is easy to converge.

As shown in Figure 2.d, the main framework is based on a residual structure, so that module is easy to converge [11]. The output from the right side of the model is multiplied by a value(*i.e.*, α in equation.(1)) to control its contribution to the final output. Given the input feature $\mathbf{X} \in \mathbb{R}^{B\times C\times W\times H}$, the CFE module $\mathbf{G}_{cfe}$ generate a feature of $\mathbf{Y} \in \mathbb{R}^{B\times \widehat{C}\times \widehat{W}\times \widehat{H}}$, the spatial dimensions of which are downsampled if the striding parameter set to be bigger than 1. The CFE operation can be formulated as:

$$\mathbf{Y} = \mathbf{G}_{cfe}(\mathbf{X}) = \alpha \cdot \mathbb{T}\left([\mathbb{F}_{+}(\mathbf{X}), \mathbb{F}_{-}(\mathbf{X})]\right) + t(\mathbf{X}), \tag{1}$$

where $\mathrm{t}(\cdot)$ is a short connection of striding function, $\mathbb{T}(\cdot)$ is the bottleneck output layer for tuning channels. To be more specific, function $\mathbb{F}$ contains four continuous operations: a 1x1 Conv layer for tuning internal channels, two adjacent 1xk and kx1 Conv layers and a 3x3 Conv layer. $\mathbb{F}_{+}(\cdot)$ and $\mathbb{F}_{-}(\cdot)$ both contain these convolution layers, the main difference of them is the ordering between 1xk Conv and kx1 Conv. As a whole, $\mathbf{G}_{cfe}$ function combines the advantages of residual learning and dual learning, and benefit to convergence in training process.

The motivation of designing function $\mathbb{F}$ is to learn more non-linearity relations, we apply the idea of split-transform-merge. In detail, splicing a 1x1 Conv layer before a larger-kernel Conv layer(initially, a k×k Conv layer), and finally followed by a 1x1 Conv layer to adjust the output dimensions. Note that the inside Conv layer with large kernel is the most significant component for feature learning. To lighten the weight and keep receptive field, we use a combination of 1xk and kx1 and its inverse combination to learn more comprehensive features. Obviously, increasing k will expand the receptive field.

After having strengthened the high-level semantics and expanded the receptive field through Conv layers, we also need to consider the problem of weights redundancy [31], which fails to achieve fully training on small-scale dataset, *e.g.*, *Pascal VOC*. Specifically, we should avoid excessive feature redundancy among channels. We use group convolution to increasing sparsity as well as shrinking the redundancy [34], which is proved to be effective by experiment. In order to assure powerful learning ability, we add another 3x3 Conv layer to compensate.

Some other details also should be concerned with. Since a 1x1 Conv layer can be used to adjust channels, the top and bottom 1x1 Conv layers together make our module flexibly inserted into arbitrary positions. Each of these Conv layers in Figure 2 represents a group of Conv+BN+ReLU layers. In addition, the Conv layer with kernel size k has (k-1)/2 padding, so that the shape of input and output features can be unchanged. Alternatively, if we have to apply a striding operation (for a stride size bigger than 1 such as 2), we set striding size of one of the 1xk kernels to be 2, and then add another Conv layer with stride 2 at the left side shortcut of the module. All of the Conv layers are initialized using the MSRA [10] method.

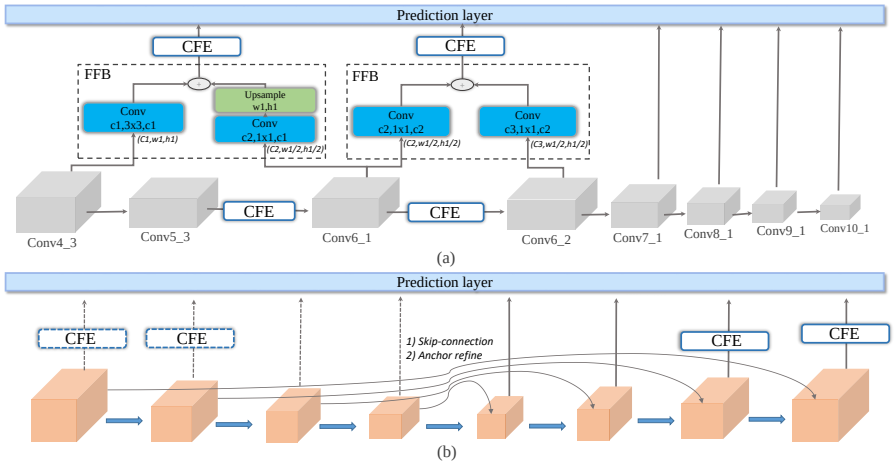

Fig. 3. (a) The way of assembling CFE modules for CFE-SSDv2. (b) The way of assembling CFE modules for DSSD(solid parts) and ReifineDet(dashed parts).

4 Detection Networks with CFE

CFE could be assembled into multiple one-stage detectors due to its flexible structure. In this section, we first introduce two architecture based on SSD, CFE-SSDv1 and v2 in section **Section 4.1**. Then we illustrate the strategy to cooperate with other detectors in **Section 4.2**, such as DSSD [5] and RefineDet [36].

4.1 CFE-SSD

CFE-SSDv1. We have analyzed how weak shallow features of SSD influence the detection results of small objects, so our target is to enhance SSD by assembling CFE modules at the most sensitive position. It's noteworthy that adding too many modules excessively is also suboptimal because this operation increase inference time as well as make training process more difficult (*i.e.*, if we broaden the input size, a smaller batch size can't fit very well). As shown in Figure 1.d, we first insert two CFE modules following the output of the backbone. Then we can get the output feature of them respectively, *i.e.*, Conv6_1 and Conv6_2 shown in Figure 1.d. These additional layers effectively increase the depth of the corresponding detection features. In addition, we connect another two separate CFE modules to Conv4_3 and Conv6_1 detection branch respectively. Since the two layers are relatively shallow, and its learned features are not high-level enough. We use CFE modules to deepen the Conv4_3 and Conv6_1 feature layers as well as broaden the receptive fields. Since this scheme only make a simple modification to the SSD architecture, we call it CFE-SSDv1. Moreover, we can find from

Figure 1.c and Figure 1.b that Conv4_3 feature indeed generates more distinguishable and position-sensitive features. The CFE modules replace the original 3x3 Conv layers in CFE-SSDv1, so this operation does not increase inference time a lot.

CFE-SSDv2. The network topology is very important for the improvement of detection accuracy. The two feature maps with the largest resolution of CFE-SSDv1 have been enhanced directly, but no other assistance has been obtained to supplement the rich semantic information from the arterial part of the network. To further improve small objects detection, we propose an advanced version, CFE-SSDv2. With the idea of feature fusion, CFE-SSDv2 improves the ability of small objects detection of shallow features compared with the CFE-SSDv1. As shown in Figure 3.a, the two shallowest feature maps are restructured through connecting feature fusion blocks (FFB) between Conv4_3 and Conv6_1, Conv6_1 and Conv6_2. Our goal is to combine a relatively shallower layer with a relatively deeper layer, which respectively represent more accurate spatial information and deeper semantic knowledge. To make models more concise and elegant, we only apply easy operations here, which also improve computing efficiency. FFB has two kinds of variants: FFB_a, shown in Figure 3.a, whose main strategy is element-wise summation; and FFB_b, where the fusion strategy is concatenation. We have compared the two kinds of blocks in experiment section 4. Due to concat operation will double the output channels, so we use the 1x1 Conv layer to slim the channels before feature fusion for fair comparison, and it leads to suboptimal compared with summation operation. Note that the FFB module is intentionally simplistic, to ensure inference efficiency and decrease training complexity. Armed with two FFB modules, CFE-SSDv2 gets more improvement. For both v1 and v2, we use hard-NMS for post-processing after filtering anchor boxes with a confidence score threshold of 0.01.

For one-stage detectors, the bottleneck of accuracy depends on backbone model and input size mostly. However, large input size (*e.g.*, 800x800) and heavy backbone (*e.g.*, SE-ResNeXt101 [13]) make the batch size in training process smaller, which will influence the model's learning ability directly and also make training time longer. According to our experience, it is required that a batch size bigger than 4 per GPU to train well an one-stage detector. So we build CFE-SSD with the consideration that compacting model structure for increasing a larger input size or changing a more powerful backbone. This is why we do not add too many extra layers for CFE-SSD.

4.2 CFE-DSSD and CFE-RefineDet

To evaluate the generalization of CFE modules, we have tested on other base detectors. DSSD [5] and RefineDet [36] are state-of-the-art one-stage detectors. They have such improvements for the original SSD. However, CFE could still benefit them. As shown in Figure 3.b, we add two CFE modules at the positions before prediction layers of the small-objects sensitive layers. In detail, the area of solid lines is for DSSD, and the area of dashed lines is for RefineDet.

We do not consider assembling more CFEs and FFBs on DSSD and RefineDet since their contribution(Encode-Decode structure, cascade regression strategy) somewhat overlaps with the effects of CFE module for helping to detect small objects. CFE modules have better performance on them, but the improvement is much smaller than on original SSD.

5 Training Details

5.1 Data augmentation

Data augmentation is essential for training. The experiments in SSD illustrate that an appropriate data augmentation strategy helps train a better model. Our strategy is the same as that of SSD [22], including image cropping, expansion, mirroring and distortion. In this work, it improves the results of CFE-SSDv1 and CFE-SSDv2 a lot especially in the term of small-size.

5.2 Design of Anchor

For a fair comparison with SSD and its variants, we identically select 6 feature layers when input size is 300×300, 7 feature layers when input size is 512×512 or 800×800, to regress detection results. As for aspect ratios of anchors, we keep the default settings of SSD except adding (3,1/3) ratio at the first regression layer when training *Pascal VOC* and *COCO*. For *DETRAC* and *BDD*, we set the parameters of anchors by analyzing the ratio distribution of ground truth. We train anchors adequately with hard negative mining strategy, which helps mitigate the extreme foreground-background class imbalance. Specifically, we select negative anchors that have top loss values to make the ratio between the negatives and positives below 3:1. Note that we judge whether a proposal is positive or not according to its biggest IoU value with ground truth. If it exceeds the threshold (0.5 as default), we treat the proposal as positive.

6 Experiments

Experiments are conducted on four datasets: *PASCAL VOC 2007*, *MS COCO*, *UA-DETRAC* and *BDD*. It's worth nothing that, for a fair comparison, we mainly compare methods based on VGG-16 backbone [29], which is pre-trained on the *ImageNet* dataset. For all experiments based on CFE modules, we start training with warm-up strategy, initialize the learning rate of 2×10^{-3}, and then drop to 2×10^{-4} and 2×10^{-5} when loss stops decreasing. The experiment on CFE-SSDv2 with ResNet-101 backbone is conducted on a machine with 2 V100 GPUs, others are conducted on a machine with 4 Titan X GPUs. Both use CUDA 8.0 and cuDNN v6.

Table 1. Detection results comparison on *PASCAL VOC07* dataset

Method	Data	Backbone	Input size	#Boxes	FPS	mAP
two-stage:						
Fast R-CNN [7]	07+12 trainval	VGG-16	∼1000 × 600	∼2000	0.5	70.0
Faster R-CNN [27]	07+12 trainval	VGG-16	∼1000 × 600	300	7	73.2
Faster R-CNN [27]	07+12 trainval	ResNet-101	∼1000 × 600	300	2.4	76.4
ION [1]	07+12 trainval	VGG-16	∼1000 × 600	4000	1.25	76.5
MR-CNN [6]	07+12 trainval	VGG-16	∼1000 × 600	250	0.03	78.2
R-FCN [3]	07+12 trainval	ResNet-101	∼1000 × 600	300	9	80.5
one-stage:						
YOLO [24]	07+12 trainval	GoogleNet	448 × 448	98	45	63.4
RON384 [14]	07+12 trainval	VGG-16	384 × 384	30600	15	75.4
SSD300* [22]	07+12 trainval	VGG-16	300 × 300	8732	46	77.2
DSOD300 [28]	07+12 trainval	DenseNet-64	300 × 300	8732	17.4	77.7
DSSD321 [5]	07+12 trainval	ResNet-101	321 × 321	17080	9.5	78.6
RefineDet320 [36]	07+12 trainval	VGG-16	320 × 320	6375	40.3	80.0
CFE-SSDv1-300	07+12 trainval	VGG-16	300 × 300	11620	42.2	80.2
CFE-SSDv2-300	07+12 trainval	VGG-16	300 × 300	11620	41.5	**80.5**
YOLOv2 [25]	07+12 trainval	Darknet-19	544 × 544	845	40	78.6
SSD512* [22]	07+12 trainval	VGG-16	512 × 512	24564	19	79.8
DSSD513 [5]	07+12 trainval	ResNet-101	513 × 513	43688	5.5	81.5
RefineDet512 [36]	07+12 trainval	VGG-16	512 × 512	16320	24.1	81.8
CFE-SSDv1-512	07+12 trainval	VGG-16	512 × 512	32756	22.0	81.8
CFE-SSDv2-512	07+12 trainval	VGG-16	512 × 512	32756	21.2	**82.1**

6.1 Results on *PASCAL VOC 2007*

In this experiment, we train our models on the union set of *VOC 2007* and *VOC 2012 trainval* set (approximately 16,500 pictures) with 20 categories and test them on the *VOC 2007 test* set. Stochastic Gradient Descent (SGD) with momentum of 0.9 and weight decay of 0.0005 was used for optimization. Table 1 shows the results.

Compare with previous state-of-the-art methods. In Table 1 , SSD300* and SSD512* are the upgraded version of SSD with the new data augmentation methods [22], which are the baselines for one-stage detectors. By integrating our CFE module, the original SSD300 model obtains a mAP of 80.5% at the speed of 42.5fps. Obviously, CFE-SSDv2 outperforms SSD and also keeps its speed. Compared with state-of-the-art two-stage methods, CFE-SSDv2 300 outperforms most of them and achieves the same accuracy as R-FCN, whose backbone is ResNet-101 and input size is $\sim 1000 \times 600$. By using a larger input size 512×512, CFE-SSDv2 512 gets 82.1% mAP, better than most object detection systems for both one-stage and two-stage. In addition, CFE-SSDv2 exceeds another version of modified SSD, DSSD [5], by a large margin. DSSD321 equipped with ResNet101 and broadening the input size to 321×321 still gets lower results by 2% compared with our method. As for input size of 512(including 513), we still outperform the heavy DSSD and also keep real-time efficiency. RefineDet is another state-of-the-art one-stage method. Both our 300(including 320) and 512 versions are higher than each of RefineDet. Specifically, CFE-SSDv2 300 not only performs better but is also more accurate. CFE-SSDv2 512 outperforms RefineDet512 as for detection accuracy and achieves state-of-the-art.

Table 2. Comparison of different modules

module name	CFE-SSD	result(mAP)
SSD(baseline) [22]	None	77.24
Inception module [31]	v1	78.91
Large separable module(k=7)[16]	v1	78.10
ResNeXt module[34]	v1	77.44
CFE module (k=3)	v1	79.96
CFE module (k=5)	v1	80.05
CFE module (k=7)	v1	**80.16**
CFE module (k=7)	v2	**80.49**

Table 3. Results of different groups in CFE(when k=3)

groups	results(mAP)
1	79.31
2	79.58
4	79.64
8	**79.96**
16	79.88
32	79.42
64	78.92

Table 4. Comparison between different CFE-SSD settings

FFB type	FFB	Result(mAP)	Speed(fps)
None (v1)	0	80.16	42.21
Concat	1	80.42	42.01
Concat	2	80.43	41.77
Sum (v2)	2	**80.49**	41.48

Table 5. Results of DSSD and RefineDet with CFE modules

Method	Size	CFEs	VOC07	COCO
DSSD	320	0	77.47	27.4
CFE-DSSD	320	2	**78.86**	**29.5**
RefineDet	320	0	80.01	29.4
CFE-RefineDet	320	2	80.00	**30.5**

Analysis of different modules. We conducted experiments to evaluate the components of CFE-SSDv1 and v2, CFE module itself and also test the generalizations of CFE module on DSSD [5] and RefineDet [36]. If not specified, default settings are: the large kernel is k=7; group convolution of 8; input size is 300×300; base architecture is CFE-SSDv1. The results are shown in Table 2∼ Table 5.

Compare with different modules. To prove that CFE does have great effects for improving detection performance, we compare CFE module with other popular modules to assemble in SSD. As shown in Table 2. CFE module with k=3 already outperforms the existing modules, and increasing k will further improve the accuracy. We have tried a larger k, but the performance begins to drop, it can be concluded that CFE module with k=7 is enough for keeping a large receptive field. The last line in Table 2 shows the effects of changing the topology structure to CFE-SSDv2.

Groups. We suggest to group convolution kernels to decrease the module weight as well as reducing the FLOPs and find that this operation can also improve detection accuracy. We deduce that group convolution controls the redundancy and even learns better feature on small-scale dataset (*e.g.*, Pascal VOC). As shown in Table 3, grouping convolution kernels to 8 groups is relatively the best.

FFB. We compare the effectiveness and efficiency of different settings of FFB, CFE-SSDv1 is relatively more efficient while CFE-SSDv2 is more accurate one that achieves a better result. Due to we keep the output channels, concat has only 1/2 input channels as summation operation. Furthermore, we select FFB type of summation(sum shown in Table 4) as the final fusion method for CFE-SSDv2.

CFE on other detectors. Moreover, not only SSD, DSSD and RefineDet could also get improvements by the assembling of CFE modules as illustrated in Table 5. The networks are illustrated in Section 4. Both of the two detectors insert two CFE modules. The improvement on *Pascal VOC* is relatively less due to the reason that different methods may contribute similarly and somewhat has increased the redundancy instead. We further evaluate it on MS-COCO 2014

Table 6. Detection accuracy comparisons in terms of mAP percentage on *MS COCO* test-dev set

Method	Backbone	Input size	Avg. Precision, IoU: 0.5:0.95	0.5	0.75	Avg. Precision, Area: S	M	L
SSD300* [22]	VGG-16	300×300	25.1	43.1	25.8	6.6	25.9	41.4
RON384++ [14]	VGG-16	384×384	27.4	49.5	27.1	-	-	-
DSSD321 [5]	ResNet-101	321×321	28.0	46.1	29.2	7.4	28.1	**47.6**
RefineDet320 [36]	VGG-16	320×320	29.4	49.2	31.3	10.0	32.0	44.4
CFE-SSDv1-300	VGG-16	300×300	29.3	49.0	31.0	10.6	31.7	44.3
CFE-SSDv2-300	VGG-16	300×300	**30.4**	**50.5**	**31.3**	**12.1**	**32.2**	46.4
YOLOv2 [25]	DarkNet-19	544×544	21.6	44.0	19.2	5.0	22.4	35.5
YOLOv3 [26]	DarkNet-53	608×608	33.0	57.9	34.4	18.3	35.4	41.9
SSD512* [22]	VGG-16	512×512	28.8	48.5	30.3	10.9	31.8	43.5
DSSD513 [5]	ResNet-101	513×513	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet500 [19]	ResNet-50	500×500	32.5	50.9	34.8	13.9	25.8	46.7
RefineDet512 [36]	VGG-16	512×512	33.0	54.5	35.5	16.3	36.3	44.3
RefineDet512 [36]	ResNet-101	512×512	36.4	57.5	39.5	16.6	39.9	51.4
CFE-SSDv1-512	VGG-16	512×512	33.8	55.1	35.7	16.2	37.2	46.3
CFE-SSDv2-512	VGG-16	512×512	35.2	56.4	36.9	17.5	38.3	47.5
CFE-SSDv2-512	ResNet-101	512×512	**39.6**	**60.3**	**42.7**	**21.2**	**44.2**	**53.1**
RefineDet320-MS[36]	VGG-16	320×320	35.2	56.1	37.7	19.5	37.2	47.0
RefineDet512-MS[36]	VGG-16	512×512	37.6	58.7	40.8	22.7	40.3	48.3
RefineDet512-MS[36]	ResNet-101	512×512	41.8	62.9	45.7	25.6	45.1	54.1
CFE-SSDv2-300-MS	VGG-16	300×300	**36.7**	**58.0**	**39.1**	**21.8**	**39.2**	**47.4**
CFE-SSDv2-512-MS	VGG-16	512×512	**39.0**	**60.5**	**41.2**	**25.0**	**42.8**	**48.5**
CFE-SSDv2-512-MS	ResNet-101	512×512	**43.1**	**63.2**	**49.2**	**28.9**	**49.1**	**56.1**

minival set and find that the improvement on both detectors is very significant compared with the corresponding original methods.

6.2 Results on *MS COCO*

Besides *Pascal VOC 07*, we also evaluate CFE-SSD on a large detection dataset, *MS COCO*. Although some module congurations may only be suitable for medium/small-scale dataset (*e.g.*, the number of groups may perform differently on larger datasets), we still keep them unchangeable in this experiment for convenience. Except VGG backbone, we also implement experiment with ResNet-101 backbone to compare with state-of-the-art detectors. Table 6 shows the result on *MS COCO* test-dev set. When equipped with VGG, our CFE-SSDv2 300 achieves 30.4% mAP and CFE-SSDv2 512 achieves 35.2% mAP. Specifically, only CFE-SSDv2 exceeds 30% of mAP of the overall detection results when input size is 300x300. And the result of small objects exceeds a large margin than other methods, which proves that our method significantly improves to detection small objects. We also compare with the popular efficient network YOLOv3 [26](19.5fps), CFE-SSDv2(16.7fps) is more accurate with comparable speed. CFE-SSDv2 512 gains 17.5% AP for small objects, 35.2 mAP for overall, which outperforms all of the other one-stage detectors. This emphasizes that the proposed CFE modules largely enhance the detection abilities of shallow layers. When changing to ResNet-101 backbone, the CFE-SSDv2 further get a large improvement. CFE-SSDv2 512 then reaches AP of 39.6 and achieve state-of-the-art results among one-stage detectors, and it also has a speed of 11 fps. In the bottom 6 lines of Table 6, we compare performance with multi-scale inference strategy, both

Table 7. Comparison between CFE-SSD and state-of-the-art methods on *DETRAC* dataset

Method	Input size	Overall	Easy	Medium	Hard	Cloudy	Night	Rainy	Sunny	FPS
YOLOv2 [25]	544×544	57.72	83.28	62.25	42.44	57.97	64.53	47.84	69.75	40
LateralCNN	1000×600	67.25	89.56	73.59	51.61	69.11	74.36	55.77	78.66	23.4
SSD[22]	512×512	72.56	91.12	77.71	57.83	79.12	73.52	59.03	81.72	22.4
RefineDet [36]	512×512	76.38	92.60	83.05	63.03	82.84	78.92	64.19	**87.32**	21.1
CFE-SSDv1	512×512	78.72	93.58	84.85	66.15	**84.72**	80.40	69.43	86.97	21.0
CFE-SSDv2	512×512	**79.25**	**94.31**	**84.93**	**67.37**	84.35	**80.60**	**70.42**	87.24	20.7
CFE-SSDv2*	512×512	**82.68**	**94.60**	**89.71**	**70.65**	**89.81**	**83.02**	**73.35**	**88.11**	-

scales of 300 and 512 or VGG and ResNet-101 backbones, CFE-SSDv2 outperforms RefineDet. Notably, The CFE-SSDv2 with multi-scale inference strategy has achieved mAP of 43.1, which is the best accuracy result of one-stage detectors. To demonstrate the targeted improvement of our model, we bolded the best result in Table 6.

6.3 Results on *UA-DETRAC*

UA-DETRAC [33] is a challenging real-world multi-object detection and multi-object tracking benchmark. The dataset contains 10 hours of videos captured with a Cannon EOS 550D camera at 24 different locations in China. The videos are recorded at 25 fps, with a resolution of 960x540 pixels. Annotation contains 4 categories in total*(car, van, bus, others)*. The difficulty behind this dataset is small and densely distributed cars at varying weather and illuminations.

As shown in Table 7, *UA-DETRAC* evaluates comprehensive detection results, including Overall, Easy, Medium, Hard, Cloudy, Night, Sunny and Rainy. The result can reflect the complete effectiveness of object detectors. As for comparisons, we select state-of-the-art one-stage detectors, such as SSD and RefineDet. Specifically, CFE-SSD exceeds raw SSD by about 7%, and exceeds RefineDet nearly by 3%. Our CFE-SSD nearly wins every condition, especially outperforming RefineDet by 4% for hard targets detection. In addition, multi-scale inference strategy make CFE-SSDv2 512 improves to 82.6% finally, ranking first place at the leaderboard. It's noteworthy that the results of RefineDet and SSD are trained and tested by ourselves, while others are from the leaderboard.

6.4 Results on *Berkeley DeepDrive*

BDD [35] is a well-annotated dataset that includes road object detection, instance segmentation, driveable area segmentation and lane markings detection annotations. The road object detection contains 2D bounding boxes annotated on 100,000 images for bus, traffic light, traffic sign, person, bike, truck, motor, car, train, and rider, 10 categories in total. The split ratio of training, validation and testing set is 7:1:2. The evaluated IoU threshold is 0.7 on testing leaderboard.

In experiment, we mainly compare CFE-SSDv2 with original SSD and RefineDet, shown in Table 8. The version with input size of 512 could realize real-

Fig. 4. Qualitative result examples. (a) Images from *COCO*, (b) Images from *VOC07*, (c) Images from *UA-DETRAC*, (d) Images from *BDD*.

time performance while The version with input size of 800 with multi-scale inference strategy achieves the second place of the challenge on the leaderboard. Note that, to evaluate the effects of small object detection, we average the results of *traffic light* and *traffic sign*(denoted by *S-mAP* in Table 8) for comparison.

Table 8. AP and FPS Results on BDD

Method	Input size	FPS	mAP(%)	S-mAP(%)
SSD	512 × 512	**23.1**	14.1	9.2
RefineDet [36]	512 × 512	22.3	17.4	13.1
CFE-SSDv2	512 × 512	21.0	**19.1**	**15.4**
CFE-SSDv2	800 × 800	6.2	**22.3**	**18.1**
CFE-SSDv2*	800 × 800	-	**29.7**	**26.0**

7 Conclusions

In this paper, we propose an effective and lightweight feature enhancing module, *i.e.*, CFE module, to improve the detection accuracy of the one-stage detectors. By assembling this module, the performance of the state-of-the-art one-stage detectors can be significantly improved while their efficiency nearly doesn't drop. Specifically, by inserting 4 CFE modules into the network architecture of the most widely used one-stage detector SSD, we get two novel one-stage detectors CFE-SSDv1 and CFE-SSDv2, which have significantly better performance than the original SSD. Experimental results on *PASCAL VOC07* and *MS COCO* datasets show that the CFE-SSDv2 outperforms state-of-the-art methods DSSD and RefineDet, especially for the small objects. Additional ablation study on *PASCAL VOC07* dataset demonstrates the effectiveness of the proposed CFE module. Moreover, CFE-SSDv2 achieves the best result on the *UA-DETRAC* dataset for vehicle detection and ranks second on the *BDD* leaderboard for road object detection, which demonstrate that it is effective for practical applications such as traffic scene video surveillance and autonomous driving.

References

1. Bell, S., Zitnick, C.L., Bala, K., Girshick, R.B.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: CVPR 2016. pp. 2874–2883 (2016)
2. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: CVPR 2017. pp. 1800–1807 (2017)
3. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. NIPS (2016)
4. Everingham, M., Gool, L.J.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes (VOC) challenge. International Journal of Computer Vision **88**(2), 303–338 (2010)

5. Fu, C., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: DSSD : Deconvolutional single shot detector. CoRR **abs/1701.06659** (2017)
6. Gidaris, S., Komodakis, N.: Object detection via a multi-region and semantic segmentation-aware CNN model. In: ICCV 2015. pp. 1134–1142 (2015)
7. Girshick, R.B.: Fast R-CNN. In: ICCV 2015. pp. 1440–1448 (2015)
8. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR 2014. pp. 580–587 (2014)
9. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. ICCV (2017)
10. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: ICCV 2015. pp. 1026–1034 (2015)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR 2016. pp. 770–778 (2016)
12. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR **abs/1704.04861** (2017)
13. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. CVPR (2018)
14. Kong, T., Sun, F., Yao, A., Liu, H., Lu, M., Chen, Y.: RON: reverse connection with objectness prior networks for object detection. In: CVPR 2017. pp. 5244–5252 (2017)
15. Kong, T., Yao, A., Chen, Y., Sun, F.: Hypernet: Towards accurate region proposal generation and joint object detection. In: CVPR 2016. pp. 845–853 (2016)
16. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J.: Light-head R-CNN: in defense of two-stage object detector. CoRR **abs/1711.07264** (2017)
17. Lin, M., Chen, Q., Yan, S.: Network in network. ICLR (2014)
18. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR 2017. pp. 936–944 (2017)
19. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV 2017. pp. 2999–3007 (2017)
20. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: ECCV 2014. pp. 740–755 (2014)
21. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. CVPR (2018)
22. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C.: SSD: single shot multibox detector. In: ECCV 2016. pp. 21–37 (2016)
23. Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: Large kernel matters - improve semantic segmentation by global convolutional network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 1743–1751 (2017)
24. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR 2016. pp. 779–788 (2016)
25. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: CVPR 2017. pp. 6517–6525 (2017)
26. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv (2018)
27. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Annual Conference on Neural Information Processing Systems 2015. pp. 91–99 (2015)
28. Shen, Z., Liu, Z., Li, J., Jiang, Y., Chen, Y., Xue, X.: DSOD: learning deeply supervised object detectors from scratch. In: ICCV 2017. pp. 1937–1945 (2017)

29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
30. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR 2015. pp. 1–9 (2015)
31. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR 2016. pp. 2818–2826 (2016)
32. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. International Journal of Computer Vision **104**(2), 154–171 (2013)
33. Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M., Qi, H., Lim, J., Yang, M., Lyu, S.: UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. arXiv CoRR **abs/1511.04136** (2015)
34. Xie, S., Girshick, R.B., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR 2017. pp. 5987–5995 (2017)
35. Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., Darrell, T.: BDD100K: A diverse driving video database with scalable annotation tooling. CoRR **abs/1805.04687** (2018)
36. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. In: CVPR (2018)