Proceedings of Machine Learning Research 101:236–251, 2019 ACML 2019

SPCDet: Enhancing Object Detection with Combined Feature Fusing

Haixin Wang HAIXINWA@BIT.EDU.CN
Beijing Institute of Technology, Beijing, China

Lintao Wu LINTAOW.LY@GMAIL.COM
Intel Labs China, Beijing, China

Qiongzhi Wu WQZ_BIT@BIT.EDU.CN
Beijing Institute of Technology, Beijing, China

Abstract

Feature pyramid and feature fusing are widely used in object detection. Using feature pyramid can confront the challenge of scale variation across different objects. Feature fusing imports context information to improve detection performance. Although detecting with feature pyramid and feature fusing has achieved some encouraging results, there are still some limitations owing to the features' level variance among different layers. In this paper, we exploit that serial-parallel combined feature fusing can enhance object detection. Instead of detecting on the feature pyramid of backbone directly, we fuse different layers from backbone as base features. Then the base features are fed into a U-shape module to build local-global feature pyramid. At last, we use the pyramid to do the multi-scale detection with our combined feature fusing method. We call this one-stage detector SPCDet. It keeps real time speed and outperforms other detectors in trade-off between accuracy and speed.

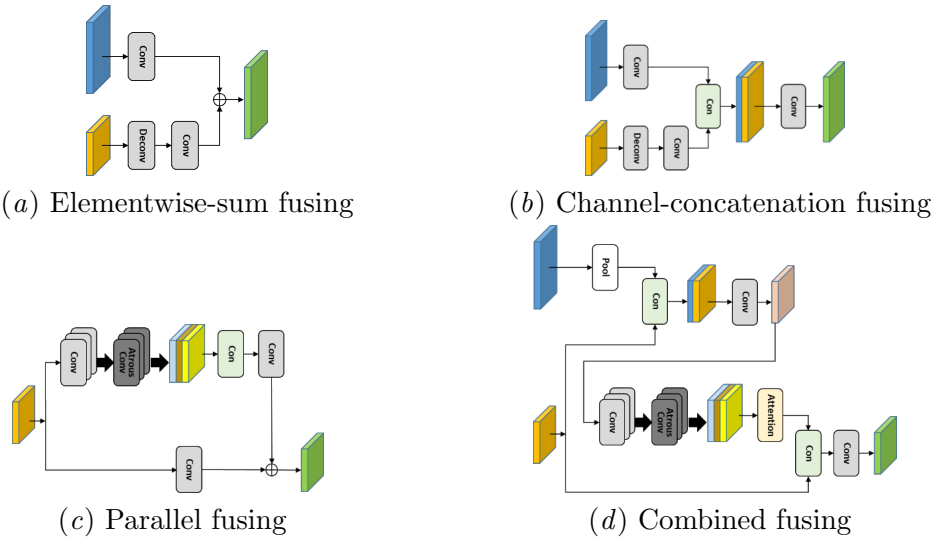
Keywords: Real-time Object Detection; Combined Feature Fusing; Context

1. Introduction

Current state-of-the-art object detectors include one-stage approaches and two-stage approaches. Two-stage approaches have better precision but lower speed. By contrast, one-stage approaches have higher speed but lower precision. To exploit a real-time high-performance object detector, many modified methods are presented base on one-stage approach, such as YOLO Redmon et al. (2016) and SSD Liu et al. (2016). It is well accepted that scale variation across object instances is one of the major challenges He et al. (2015), Lin et al. (2017a). To solve this problem, feature pyramid and feature fusing are widely used in one-stage object detector Liu et al. (2016), Lin et al. (2017a).

Context information plays an important role in object detection task, especially for small objects Cao et al. (2017). On the one hand, the context fusing collects relative information that helps to recognize objects which have poor semantic features. On the other hand, contextual features combine semantic and appearance information, which makes features robust to do both classification and regression tasks. Moreover, the fusing of contexts can weaken the background noises. To fuse context in different stages, elementwise-sum fusing

in Fig. 1(a) Cao et al. (2017) and channel-concatenation fusing in Fig. 1(b) Cao et al. (2017), Fu et al. (2017) are usually used. To produce features in different receptive field sizes for context fusing, the parallel fusing method in Fig. 1(c) Chen et al. (2017), Dai et al. (2017), Liu et al. (2018) uses multi-branches which include different atrous convolution kernels. In our paper, we use a combined fusing method in Fig. 1(d) which includes stage-wise fusing and receptive-field-wise fusing. This method can get richer context information.



(a) Elementwise-sum fusing

(b) Channel-concatenation fusing

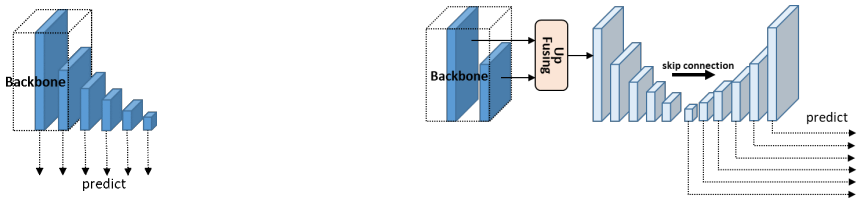
(c) Parallel fusing

(d) Combined fusing

Figure 1: Four typical structures of feature fusing. (a) Two feature maps in different resolutions are fused by element-sum operation. The low resolution maps are upsampled by deconvolution. (b) This module is similar to figure 1(a), but the two maps are concatenated in channel dimension. (c) Using parallel convolution kernels with different atrous rates can get maps in different receptive fields. This structure is similar to inception Szegedy et al. (2015), Szegedy et al. (2016), Szegedy et al. (2017). (d) This module contains serial down fusing and parallel fusing. The serial down fusing down-samples high-resolution maps and concatenates them with low-resolution maps. The parallel fusing is similar to Fig.1(c), but a channel-wise attention is inserted into the module.

In object detection task, backbone of detector is usually pre-trained on classification task. It is not appropriate to set the layers in backbone as the detection pyramid directly. Classification task concentrates on semantic information. But for detection task, local texture information is critical. In addition, classification task needs more down-sampling operations, but detection task needs shallow layers to predict small object. In Fig. 2(a), Feature pyramid of SSD Liu et al. (2016) contains 2 layers in backbone (conv4_3, conv7) and 4 extra layers. There are two issues: 1) Generally, detector's backbone is pre-trained for classification task. Using features in backbone for object detection directly is not adaptive. 2) Shallow layers have pool semantic features, and deep layers have pool local appearance features. There is a dilemma between classification and regression. In Fig. 2(b), our method

uses two layers from backbone to produce base features. This avoids using backbone's feature layers directly. Then we use U-shape module to produce feature pyramid with rich context information. This structure fuses deep and shallow layers with skip connection. It builds a pyramidal feature hierarchy with both high-level semantics and local information throughout. Recently, many networks Lin et al. (2017a), Fu et al. (2017), Newell et al. (2016), Kong et al. (2017), Shrivastava et al. (2016) use the similar structure.



(a) SSD type feature pyramid (b) Our feature pyramid with rich context information

Figure 2: (a) SSD directly uses two layers of backbone (i.e.VGG16 Simonyan and Zisserman (2015)) and four extra layers to construct the feature pyramid. (b) In our method, we use the up fusing module to fuse two layers from backbone and get a base layer. Then we use U-shape module to produce feature pyramid.

In summary, our contributions are listed as follows: 1) We present a combined fusing method and a novel real-time one-stage detector named SPCDet. 2)We test SPCDet on MS COCO benchmark. It achieves a mAP of 33.2 at speed of 40 FPS and a mAP of 37.3 at speed of 21.4 FPS in 320×320 resolution and 512×512 resolution respectively on COCO test-dev. It is time efficient and powerful.

2. Related Work

Object detection is a topic of general interest in computer vision. Almost all the object detectors are based on deep convolutional networks recently. Current state-of-the-art object detectors include two types of fundamental framework. The first one is two-stage framework, such as Faster RCNN Ren et al. (2015) and FPN Lin et al. (2017a). These methods first produce proposal regions as the object candidates, and then did proposal classification and bounding box regression for each candidate. Due to the speed limit of two-stage approaches, a number of researches focus on one-stage framework. Some representative algorithms are YOLO and SSD. These object detectors directly do classification and bounding box regression on convoluted dense feature maps without proposal candidates.

Feature pyramid: In earlier algorithms, such as Faster RCNN and YOLO, single feature layer is used for object predicting. It is not appropriate for these anchor-based methods. Because in single feature layer, mismatching between anchor scale and receptive field makes tiny and huge objects hard to be detected. In later work, SSD uses feature pyramid as the multi-scale representation. It sets default boxes of different scales on different output layers and uses shallower feature layers to predict smaller objects, while uses deeper feature layers

to predict bigger objects. In our method, we also follow this manner to predict objects in different scales. But the feature pyramid is not directly made up of backbone's layers. We use pyramid from U-shape module which has rich context information.

U-shape module: U-shape module is like a hourglass structure. It is widely applied in image recognition tasks, such as pose estimation Newell et al. (2016), semantic segmentation Badrinarayanan et al. (2017), Paszke et al. (2016), and object detection Lin et al. (2017a), Fu et al. (2017). The U-shape module includes a bottom-up pathway, a top-down pathway and lateral skip connections. The bottom-up pathway is the feed forward convolution computation which computes a feature hierarchy consisting of feature maps at several scales with stride larger than one. The top-down pathway up-samples features or uses deconvolution computation to produce higher-resolution features which are spatially coarser but semantically stronger. The lateral skip connections merges feature maps of the same resolution from the bottom-up pathway and the top-down pathway. This operation produces feature maps with rich contexts which include both localized and global information.

Context fusing: Context information is effective for accuracy improvement in object detection. It is used in many previous algorithms, such as ION Bell et al. (2016), Feature-fused SSD Cao et al. (2017), FPN, ASPP Chen et al. (2017), RFBNet Liu et al. (2018). In these algorithms, classification and regression results are not predicted from single feature layer, but from fused feature layers. There are generally two strategies for context fusing. One is fusing feature maps from different stages. Because of the different resolution, deconvolution is often used in fusing method. Another one is fusing features from different receptive field. Multi-branch atrous convolution computation is a common method to get features in different receptive fields. In our paper, we present a novel fusing method which combines two strategies mentioned above. This method can get richer context information. And in our knowledge, we are the first one to present this combined method.

3. Method

The framework of SPCDet is shown in Fig. 3. SPCDet uses backbone to extract row feature from image. Then it fuses two layers from backbone to generate base features. The base features are fed into U-shape module to produce feature pyramid which indicates instances in different scales. Before feeding pyramid into detection convolution layers, SPCDet uses a combined fusing method to generate robust features. Similar to SSD, dense bounding boxes and category scores are generated in multiple scales. There is a score threshold to remove almost low-score boxes. Then the nms (Non-Maximum Suppression) operation filters out redundant boxes. In this paper, we use GPU-based nms and CPU-based soft-nms Bodla et al. (2017) with a linear kernel.

3.1. Future Pyramid

Base Feature: To get base feature maps that is not from backbone directly, the up fusing module shown in Fig. 4(a) uses convolution, deconvolution computations and concatenation operation to fuse features from backbone. As a base layer, base feature should have an

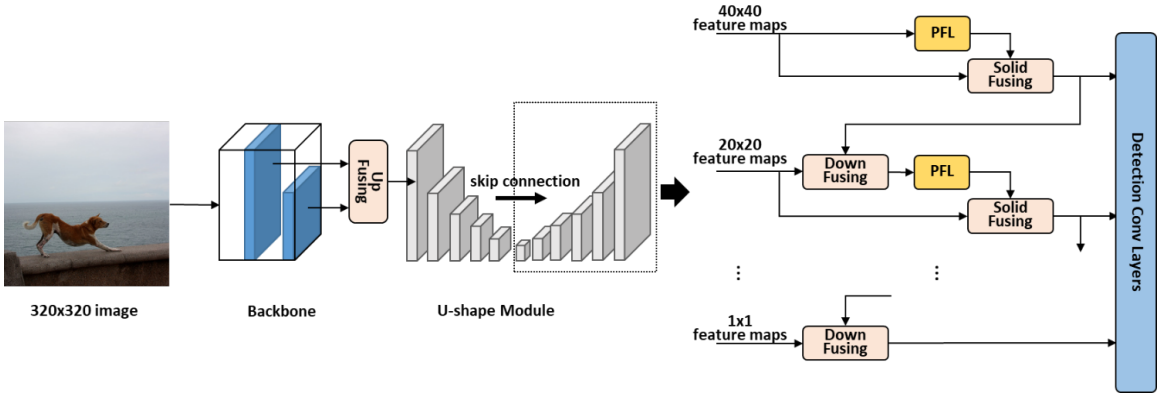


Figure 3: Overview of SPCDet (input size is 320×320). Instead of using maps in backbone as the pyramid directly, SPCDet fuses 16-stride and 8-stride maps to get 40×40 base feature maps and feeds them into the U-shape module to produce a pyramid which contains 6 layers. Every layer in pyramid is fed into the combined fusing method to generate features with context information. At last, the features are fed into detection convolution layers.

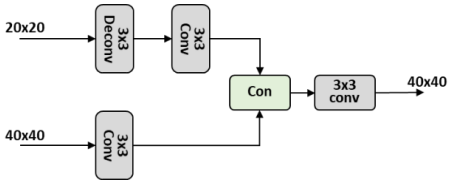
appropriate resolution and multiple feature levels. So, 8-stride and 16-stride feature maps from backbone are chose to be fed into up fusing module and the module outputs 8-stride base feature maps.

Feature pyramid: As shown in Fig. 4(b), the U-shape module produces a set of symmetrical feature maps in bottom-up pathway and top-down pathway. To predict object instances of different scales, this module generates 6 layers in different resolutions on both two pathways. Feature layers in top-down pathway have both local and global features and is set as the feature pyramid.

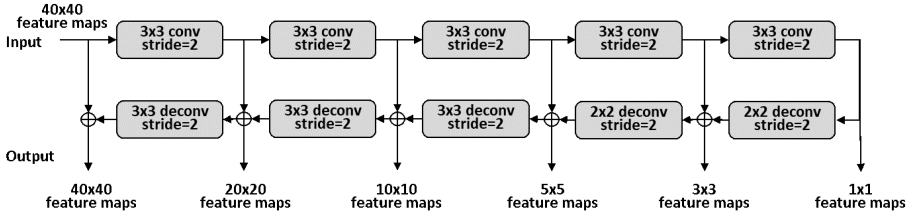
3.2. Serial-parallel Combined Fusing

The framework of serial-parallel combined fusing module is shown in Fig. 5(a). It has three pathways including serial down fusing, parallel fusing and solid fusing. In serial down fusing pathway, to get stage-wise context information, the feature maps in higher resolution are fed into pooling layer and then are concatenated with maps in lower resolution. After a convolution computation, the features are fed into parallel fusing in Fig. 5(b). The parallel fusing layer (PFL) uses multi-branch convolutions to produce feature maps in different receptive fields and then fuses them. As shown in Fig. 5(c), solid fusing pathway concatenates primitive features maps and rich-context maps from parallel fusing pathway. Before concatenating rich-context maps, the channel-wise attention Yu et al. (2018) selectively enhances useful context features and suppress less useful ones.

The combined fusing module aims to assemble context information from different scales



(a) Up fusing module



(b) U-shape module

Figure 4: (a) The up fusing module is the same as channel-concatenation fusing in Fig. 1(*b*). (b) The kernel size of convolution and deconvolution is 3×3 and the stride is 2. The shallow layers are connected with deep layers by element-sum operation. All the layers are normalized with batch normalization.

and receptive fields. The feature pyramid which is fed into this module can be presented as $X = [X_1, X_2, \ldots, X_n]$, where $X_n \in R^{W_n \times H_n \times C}$ is the feature maps of the nth scale. In the same way, the output of this module can be presented as $Y = [Y_1, Y_2, \ldots, Y_n]$, where $Y_n \in R^{W_n \times H_n \times C}$. Y_n is fed into detection convolution layer to produce dense candidate boxes. In serial down fusing phase, we can get feature maps Z_n which accommodates context information from different scales:

$$Z_n = F_{down}(X_n, Y_{n-1}) \tag{1}$$

And Z_n is fed into PFL to generate context in multiple receptive fields:

$$\widetilde{Z}_n = F_{pfl}(Z_n) \tag{2}$$

where $\widetilde{Z}_n = [\widetilde{Z}_n^1, \widetilde{Z}_n^2, \ldots, \widetilde{Z}_n^m], m \in [1, 4]$. It means output has 4 kinds of receptive fields. Then the attention module selectively enhances useful context features:

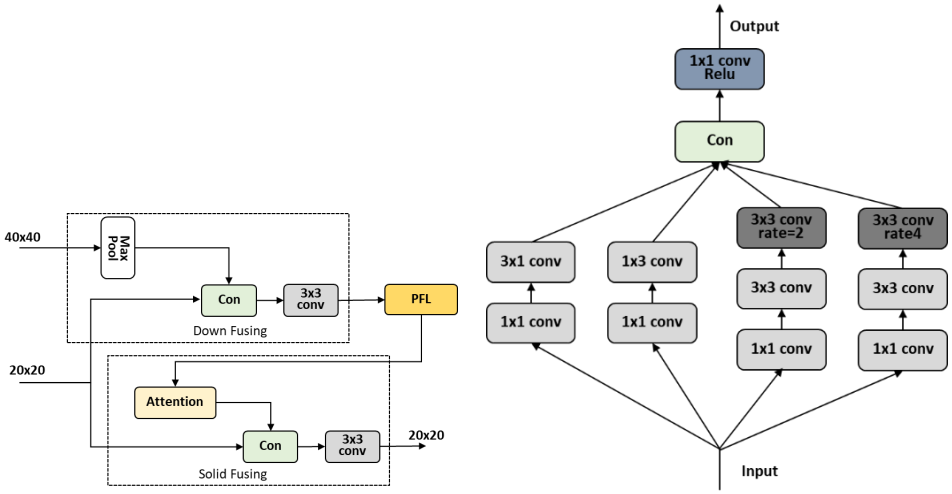
$$S_n = \widetilde{Z}_n \times W_n \tag{3}$$

where $W_n = F_{atten}(\widetilde{Z}_n)$ and $W_n \in R^{1 \times 1 \times C}$. At last, we get:

$$Y_n = F_{solid}(X_n, S_n) \tag{4}$$

3.3. Powerful Backbone

We know that ResNet He et al. (2016) is deeper than VGG-16 and has excellent performance on ImageNet's classification task. To get higher level of semantic, it needs more



(a) Serial-parallel combined fusing (b) Detail of parallel fusing layer

(c) Channel-wise attention

Figure 5: (a) Serial-parallel combined fusing module contains serial down fusing, parallel fusing and solid fusing. (b) Similar to inception structure, PFL has multiple branches. It uses 1×1 convolution to reduce channel numbers and uses different convolution kernels to get maps in various receptive fields. Then the maps are fused by concatenation and 1×1 convolution (c) Channel-wise attention uses average pooling to squeeze layer's width and height into 1×1. Then it uses two 1×1 convolutions to do non-liner shift to produce a set of channel weights. The weights activate primitive feature layers by element-wise multiplication.

down-sampling operations and concentrates more weight parameters on deeper layers. For example, ResNet-101 sets 23 bottleneck blocks in conv4_x (stride=16). This means conv4_x has 69 layers which accounts for about 70% of the total layers.

To detect small object, the stride of our detector's base feature is 8. The 16-stride layer in backbone is fused with 8-stride layer. To get better small object detection competence, we need to concentrate more weight parameters on 8-stride layers. Therefore, we change the stride of second layer's 3×3 max pooling layer from 2 to 1 in ResNet. In this way, conv4_x in ResNet has a stride of 8. But this operation results in more floating-point computations because 8-stride layer has larger resolution. In a trade-off way, we choose ResNet-50 which has 1-stride max pooling layer as the backbone. This ResNet-50 is named ResNet-50* in our paper.

4. Experiment

We train our novel detector on MS COCO Lin et al. (2014) and get comprehensive experimental results. On MS COCO, we use the trainval35k Bell et al. (2016) set for training, which is consist of 80k images from train split and a random 35k subset from the 40k-image val split. To compare with past state-of-the-art detectors, we use test-dev split which has no public labels and requires the use of the evaluation server as the test set. We also report the results of ablation studies evaluated on the minival split which includes 5000 images for convenience.

4.1. Training Detail

We use training strategy similar to that of SSD, including setting of anchors, method of data augmentation, hard negative mining and loss function. Model's backbone is pre-trained on ILSVRC CLS-LOC dataset Russakovsky et al. (2015). We set the batch size at 64 in 320×320 resolution and 32 in 512×512 resolution because of the limitation of GPU memory. The total training epochs is 150. At the start of model training, we use warm-up strategy which initializes the learning rate as 1×10^{-6} and gradually ramps up the learning rate to 4×10^{-3} at the first 5 epochs. After warm-up phase, the learning rate is divided by 10 at 90, 120, 140 epochs. Following Liu et al. (2016), we utilize a weight decay of 0.0005 and a momentum of 0.9.

4.2. Comparison with State-of-the-art

We compare experimental results of the proposed SPCDet with past state-of-the-art one-stage detectors on COCO test-dev. The experimental results are shown in Table 1. We can see that SPCDet outperforms other one-stage detectors in same resolution. Especially in small objects, our SPCDet leads other detectors a lot. The AP^s at 20.4 of SPCDet512-ResNet-50* is close to some models tested in 1280×800 resolution. The references in Table 1 are as following: [1] Ren et al. (2015), [2] Zhang et al. (2018a), [3] He et al. (2016), [4] Lin et al. (2017a), [5] Shrivastava et al. (2016), [6] He et al. (2017), [7] Cai and Vasconcelos (2018), [8] Liu et al. (2016), [9] Fu et al. (2017), [10] Zhang et al. (2018b), [11] Zhang et al. (2018a), [12] Liu et al. (2018), [13] Redmon and Farhadi (2018), [14] Lin et al. (2017b).

4.3. Ablation Experiments

In order to verify the effectiveness our fusing method, we do experiments with different subcomponent combinations. All the experiments are based on 320×320 resolution and tested on COCO minival split. The result is shown in Table2.

Base: To emphasize the importance of our fusing method, we set the model which excludes serial-parallel combined fusing model but remains U-shape pyramid. This model's mAP is set as the baseline. **Serial Down Fusing & Parallel Fusing Layers:** The combined fusing strategy consists of serial and parallel structures. Our serial fusing goes across two adjacent layers in pyramid. The parallel fusing is multi-branch combination in one resolution. **ResNet-50*:** To concentrate more weight parameters on 8-stride layers of ResNet-50, we change max pooling's stride from 2 to 1 in second layer. This operation makes 8-stride

Method	Input size	Backbone	AP	AP^{50}	AP^{75}	AP^s	AP^m	AP^l
two stage								
Faster R-CNN[1]	1000×600	VGG-16	24.2	45.3	23.5	7.7	26.4	37.1
CoupleNet[2]	1000×600	ResNet-101	34.4	54.8	37.2	13.4	38.1	50.8
Faster R-CNN+++ [3]	1000×600	ResNet-101	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w/FPN [4]	1000×600	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN w/TDM [5]	1000×600	Inception-ResNet-v2	36.8	57.7	39.2	16.2	39.8	52.1
Mask R-CNN [6]	1280×800	ResNeXt-101	39.8	62.3	43.4	22.1	43.2	51.2
Cascade R-CNN [7]	1280×800	ResNet-101	42.8	62.1	46.3	23.7	45.5	55.2
one stage								
SSD300* [8]	300×300	VGG-16	25.1	43.1	25.8	6.6	25.9	41.4
DSSD321 [9]	321×321	ResNet-101	28.0	46.1	29.2	7.4	28.1	47.6
DES300 [10]	300×300	VGG-16	28.3	47.3	30.3	10.9	31.8	43.5
RefineDet320 [11]	320×320	VGG-16	29.4	49.2	31.3	10.0	32.0	44.4
RefineDet320 [11]	320×320	ResNet-101	32.0	51.4	34.2	10.5	34.7	50.4
RFBNet300 [12]	300×300	VGG-16	30.3	49.3	31.8	11.8	31.9	45.9
SPCDet320(ours)	320×320	VGG-16	**33.2**	52.2	35.8	14.0	36.6	47.2
	320×320	ResNet-50*	**34.2**	53.7	37.0	15.6	39.2	47.5
SSD512* [2]	512×512	VGG-16	28.8	48.5	30.3	10.9	31.8	43.5
DES512 [10]	512×512	VGG-16	32.8	53.2	34.6	13.9	36.0	47.6
YOLOv3 [13]	608×608	DarkNet-53	33.0	57.9	34.4	18.3	35.4	41.9
DSSD513 [9]	513×513	ResNet-101	33.2	53.3	35.2	13.0	35.4	51.1
RefineDet512 [11]	512×512	VGG-16	33.0	54.5	35.5	16.3	36.3	44.3
RefineDet512 [11]	512×512	ResNet-101	36.4	57.5	39.5	16.6	39.9	51.4
RFBNet512-E [12]	512×512	VGG-16	34.4	55.7	36.4	17.6	37.0	47.6
RetinaNet500 [14]	832×500	ResNet-101	34.4	53.1	36.8	14.7	38.5	49.1
SPCDet512(ours)	512×512	VGG-16	**37.3**	56.7	40.6	19.5	42.4	49.7
	512×512	ResNet-50*	**38.1**	58.4	41.2	20.4	42.9	51.2

Table 1: Detection accuracy comparisons on MS COCO test-dev. For fair comparison, all results are tested with single-scale inference strategy.

layers are in conv4_x which has maximum number of layers.

As can be seen from Table2, serial down fusing improves the mAP by 0.8, which shows the effectiveness of stage-wise context. PFL improves the mAP by 1.7, which shows the effectiveness of receptive-field-wise context. The combined fusing improves the mAP by 2.2, which is more powerful than single fusing method. In many object detection method, noticeable AP gain can be gotten from deeper backbone. We replace VGG16 with ResNet-50*, which improves mAP from 33.0 to 33.9.

4.4. Inference Time

We test the inference speed of SPCDet and other state-of-the-art methods on COCO test-dev. Our test machine contains CPU of Intel Core i7-8700K and GPU of NVIDIA Titan X. Our program is based on pytorch framework with python3.6 and CUDA 8.0. We set the batch as 1 and test each image's inference time. Then we save inference time of the fastest top 99% ones and calculate the average time at last.

Component	mAP				
Base	√	√	√	√	√
Serial down fusing		√		√	√
Parallel fusing with PFL			√	√	√
ResNet-50*					√
AP	30.8	31.6	32.5	33.0	**33.9**
AP^{50}	50.6	50.5	51.5	52.1	**52.9**
AP^{75}	32.7	33.5	34.9	35.6	**36.7**
AP^{s}	13.6	13.6	14.2	14.4	**15.7**
AP^{m}	35.1	35.9	36.7	37.5	**39.7**
AP^{l}	44.1	46.2	47.0	48.0	**48.7**

Table 2: Ablation result evaluated on COCO minival.

Resolution		320×320	512×512
Network inference time	-	16.8ms	30.3ms
Nms time	GPU-based nms	4.5ms	7.7ms
	CPU-based soft-nms	8.2ms	16.4ms
Total time	with GPU-based nms	21.3ms	38.0ms
	with CPU-based soft-nms	25.0ms	46.7ms

Table 3: Inference speed of SPCDet.

For SPCDet, we test network inference time and nms time separately. The result is shown in Table3. SPCDet processes an image in 21.3ms (46.9 FPS) and 25.0ms (40 FPS) with GPU-based nms and CPU-based soft-nms respectively in 320×320 resolution. This is a good trade-off between accuracy and speed with mAP of 32.8 (with GPU-based nms) and 33.2 (with CPU-based soft-nms). It means our detector keeps real-time speed and outstanding accuracy at the same time.

In Fig. 6, we compare SPCDet's (with backbone of VGG) inference speed with recent state-of-the-art one-stage detectors. RefineDet512* has backbone of ResNet-101 and SPCDet512* uses GPU-based nms which is faster than SPCDet512. In all of the models with mAP higher than 36%, our SPCDet has the fastest speed. For example, RefineDet512* achieves mAP of 36.4 with 118.2ms but SPCDet512* achieves higher mAP with only 38ms which is about one third of RefineDet512*. In all of the models with inference time lower than 30ms, our SPCDet has the highest mAP. In summary, SPCDet outperforms other detectors in trade-off between accuracy and speed.

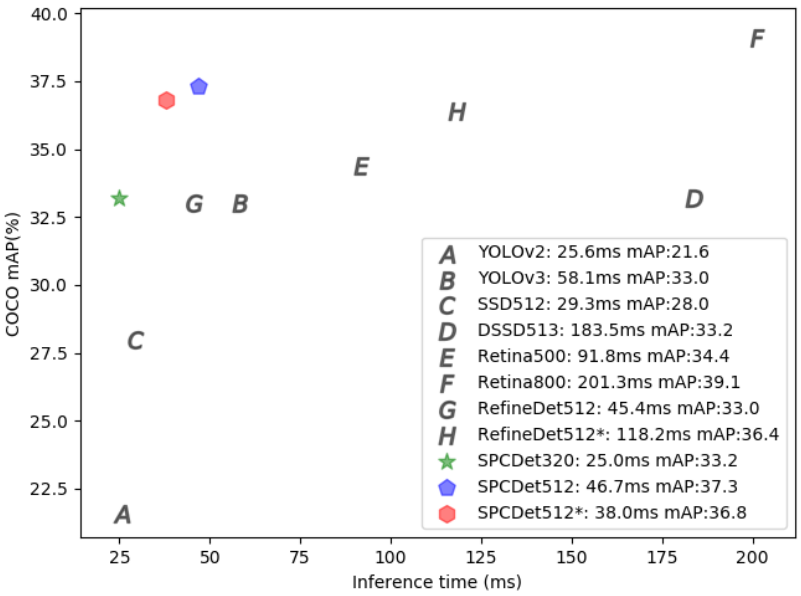


Figure 6: Speed (ms) vs. accuracy (mAP) on MS COCO test-dev.

5. Discussion

5.1. Effect of the Feature Fusing Strategy

To verify the importance of our feature fusing strategy, we visualize the activation values of base layer, U-shape layers and predicting layers (i.e. the output layers of Combined Fusing module), such an example shown in Fig. 7. A group of skiers including some tiny ones are in the input image. Features of different stages are indicated by activation values of different layers. In base layer, the background noises are obvious and submerge small objects' features. Compared with features in base layer, each object has sharper features in U-shape layers. It means base features are denoised by U-shape module. But some activation values are low. In predicting layers, objects' activation is stronger than that in U-shape layers. This benefits from the fusion of more contexts. Through the above example, we can find that our method of context fusing has an obvious effect in weakening background noises and enhancing objects' features.

5.2. Robustness of Scale Variation

Handling scale variation across object instances is significant in general object detection. There are two challenges: 1) tiny objects have poor semantic features in small scale layer; 2) objects of the same class have different scales. Fig. 8 shows an example of detection result and its corresponding activation values of multi-scale features. The input image contains

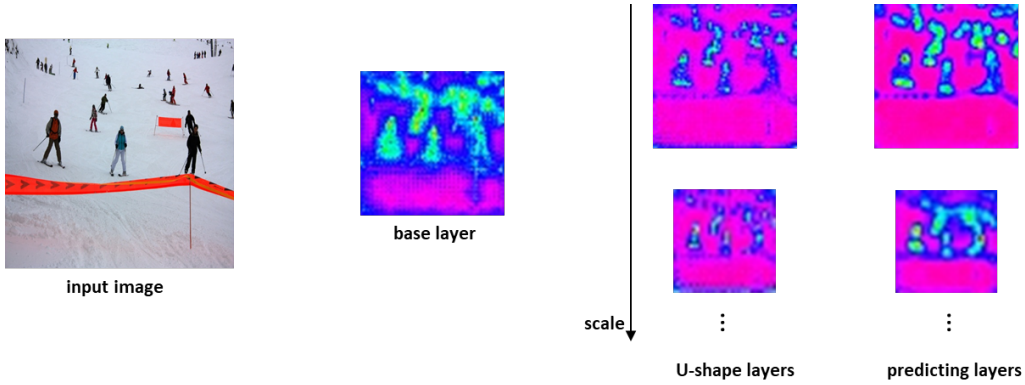


Figure 7: A case of activation values of different layers.

four people and 3 skis. Sizes among the four people are different. Two skis in image's up-left region are so small and have poor appearance information. We find that: 1) smaller person has stronger activation in maps of smaller scale, and so does bigger person; 2) even though the tiny skis almost have no appearance information, the relative contexts help them be detected by detector. These mean our detector is robust to detect objects with different scales.

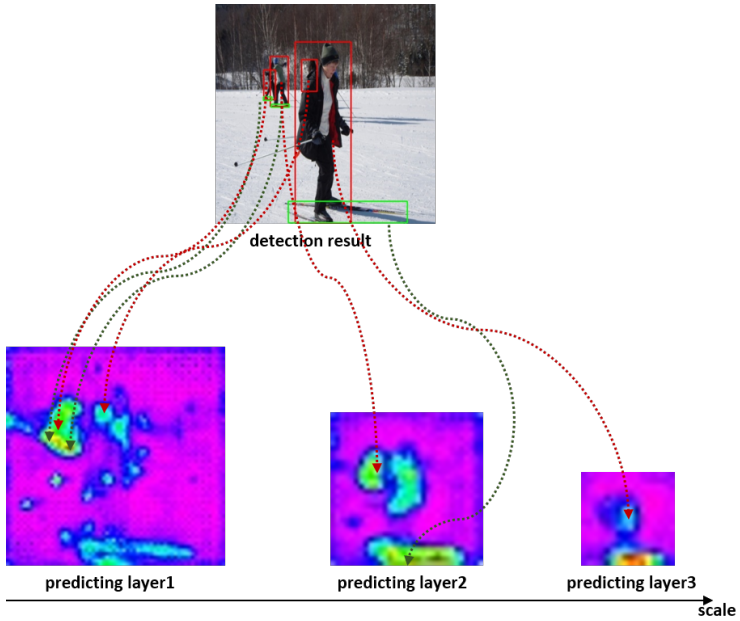


Figure 8: An example of detection result and its corresponding activation values of multi-scale features.

6. Conclusion

In this paper, we present a real-time object detector which is fast and powerful. Instead of using maps in backbone as the pyramid directly, it fuses 8-stride layer and 16-stride layer to produce base feature maps and feeds the base layer into a U-shape module to produce feature pyramid for multi-scale prediction. Then it uses a combined fusing structure to get rich context information. The combined fusing method includes serial down fusing and parallel fusing and the fusing features is selected by channel-wise attention. Our experiments show that our combined fusing method has a good performance. On COCO test-dev, our method achieves mAP of 33.2 at speed of 40.0 FPS and mAP of 37.3 at speed of 21.4 FPS in 320×320 resolution and 512×512 resolution, respectively.

References

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, 2017. doi: 10.1109/TPAMI.2016.2644615. URL https://doi.org/10.1109/TPAMI.2016.2644615.

Sean Bell, C. Lawrence Zitnick, Kavita Bala, and Ross B. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2874–2883, 2016. doi: 10.1109/CVPR.2016.314. URL https://doi.org/10.1109/CVPR.2016.314.

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-nms - improving object detection with one line of code. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 5562–5570, 2017. doi: 10.1109/ICCV.2017.593. URL https://doi.org/10.1109/ICCV.2017.593.

Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6154–6162, 2018. doi: 10.1109/CVPR.2018.00644. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Cai_Cascade_R-CNN_Delving_CVPR_2018_paper.html.

Guimei Cao, Xuemei Xie, Wenzhe Yang, Quan Liao, Guangming Shi, and Jinjian Wu. Feature-fused SSD: fast detection for small objects. *CoRR*, abs/1709.05054, 2017. URL http://arxiv.org/abs/1709.05054.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. URL http://arxiv.org/abs/1706.05587.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 764–773, 2017. doi: 10.1109/ICCV.2017.89. URL https://doi.org/10.1109/ICCV.2017.89.

Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg. DSSD : Deconvolutional single shot detector. *CoRR*, abs/1701.06659, 2017. URL http://arxiv.org/abs/1701.06659.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1904–1916, 2015. doi: 10.1109/TPAMI.2015.2389824. URL https://doi.org/10.1109/TPAMI.2015.2389824.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988, 2017. doi: 10.1109/ICCV.2017.322. URL https://doi.org/10.1109/ICCV.2017.322.

Tao Kong, Fuchun Sun, Anbang Yao, Huaping Liu, Ming Lu, and Yurong Chen. RON: reverse connection with objectness prior networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5244–5252, 2017. doi: 10.1109/CVPR.2017.557. URL https://doi.org/10.1109/CVPR.2017.557.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 740–755, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://doi.org/10.1007/978-3-319-10602-1_48.

Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 936–944, 2017a. doi: 10.1109/CVPR.2017.106. URL https://doi.org/10.1109/CVPR.2017.106.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007, 2017b. doi: 10.1109/ICCV.2017.324. URL https://doi.org/10.1109/ICCV.2017.324.

Songtao Liu, Di Huang, and Yunhong Wang. Receptive field block net for accurate and fast object detection. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI*, pages 404–419, 2018. doi: 10.1007/978-3-030-01252-6_24. URL https://doi.org/10.1007/978-3-030-01252-6_24.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, pages 21–37, 2016. doi: 10.1007/978-3-319-46448-0_2. URL https://doi.org/10.1007/978-3-319-46448-0_2.

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, pages 483–499, 2016. doi: 10.1007/978-3-319-46484-8_29. URL https://doi.org/10.1007/978-3-319-46484-8_29.

Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *CoRR*, abs/1606.02147, 2016. URL http://arxiv.org/abs/1606.02147.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. URL http://arxiv.org/abs/1804.02767.

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788, 2016. doi: 10.1109/CVPR.2016.91. URL https://doi.org/10.1109/CVPR.2016.91.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015. URL http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y. URL https://doi.org/10.1007/s11263-015-0816-y.

Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. Beyond skip connections: Top-down modulation for object detection. *CoRR*, abs/1612.06851, 2016. URL http://arxiv.org/abs/1612.06851.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.1556.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper

with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1–9, 2015. doi: 10.1109/CVPR.2015.7298594. URL https://doi.org/10.1109/CVPR.2015.7298594.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826, 2016. doi: 10.1109/CVPR.2016.308. URL https://doi.org/10.1109/CVPR.2016.308.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 4278–4284, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806.

Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1857–1866, 2018. doi: 10.1109/CVPR.2018.00199. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Yu_Learning_a_Discriminative_CVPR_2018_paper.html.

Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. Single-shot refinement neural network for object detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4203–4212, 2018a. doi: 10.1109/CVPR.2018.00442. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Single-Shot_Refinement_Neural_CVPR_2018_paper.html.

Zhishuai Zhang, Siyuan Qiao, Cihang Xie, Wei Shen, Bo Wang, and Alan L. Yuille. Single-shot object detection with enriched semantics. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 5813–5821, 2018b. doi: 10.1109/CVPR.2018.00609. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Single-Shot_Object_Detection_CVPR_2018_paper.html.