Enhancement of SSD by concatenating feature maps for object detection

Jisoo Jeong
soo3553@snu.ac.kr
Hyojin Park
wolfrun@snu.ac.kr
Nojun Kwak
nojunk@snu.ac.kr

Department of Transdisciplinary Studies
Seoul National University, Korea

Abstract

We propose an object detection method that improves the accuracy of the conventional SSD (Single Shot Multibox Detector), which is one of the top object detection algorithms in both aspects of accuracy and speed. The performance of a deep network is known to be improved as the number of feature maps increases. However, it is difficult to improve the performance by simply raising the number of feature maps. In this paper, we propose and analyze how to use feature maps effectively to improve the performance of the conventional SSD. The enhanced performance was obtained by changing the structure close to the classifier network, rather than growing layers close to the input data, e.g., by replacing VGGNet with ResNet. The proposed network is suitable for sharing the weights in the classifier networks, by which property, the training can be faster with better generalization power. For the Pascal VOC 2007 test set trained with VOC 2007 and VOC 2012 training sets, the proposed network with the input size of 300×300 achieved 78.5% mAP (mean average precision) at the speed of 35.0 FPS (frame per second), while the network with a 512×512 sized input achieved 80.8% mAP at 16.6 FPS using Nvidia Titan X GPU. The proposed network shows state-of-the-art mAP, which is better than those of the conventional SSD, YOLO, Faster-RCNN and RFCN. Also, it is faster than Faster-RCNN and RFCN.

1 Introduction

Object detection is one of the main areas of researches in computer vision. In recent years, convolutional neural networks (CNNs) have been applied to object detection algorithms in various ways, improving the accuracy and speed of object detection [11, 13, 14, 15, 17].

Among various object detection methods, SSD [11] is relatively fast and robust to scale variations because it makes use of multiple convolution layers for object detection. Although the conventional SSD performs good in both the speed and detection accuracy, it has a couple of points to be supplemented. First, as shown in Fig. 1, which shows the overall structure of conventional SSD, each layer in the feature pyramid[1] is used independently as an input to the classifier network. Thus, the same object can be detected in multiple scales. Consider a

[1] The term *feature pyramid* is used to denote the set of layers that are directly used as an input to the classifier network as shown in Fig. 1.

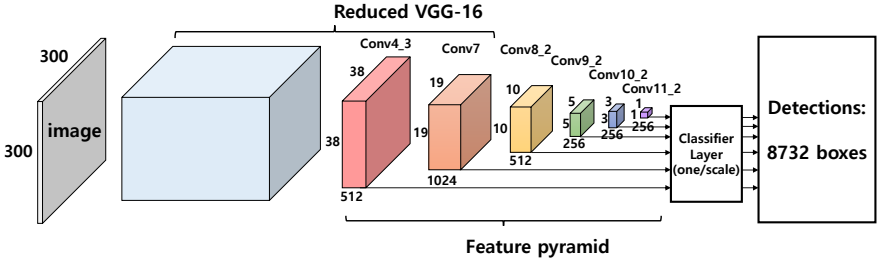


Figure 1: Overall structure of the conventional SSD. The layers from Conv4-3 to Conv11-2 which are used as the input to the classifier network is denoted as feature pyramid. Each layer in the feature pyramid is responsible for detecting objects in the corresponding size.

certain position of a feature map in a lower layer (say, Conv4-3) is activated. This information can affect entire scales up to the the last layer (Conv11-2), which means that the relevant positions in the higher layers have a good chance to be also activated.

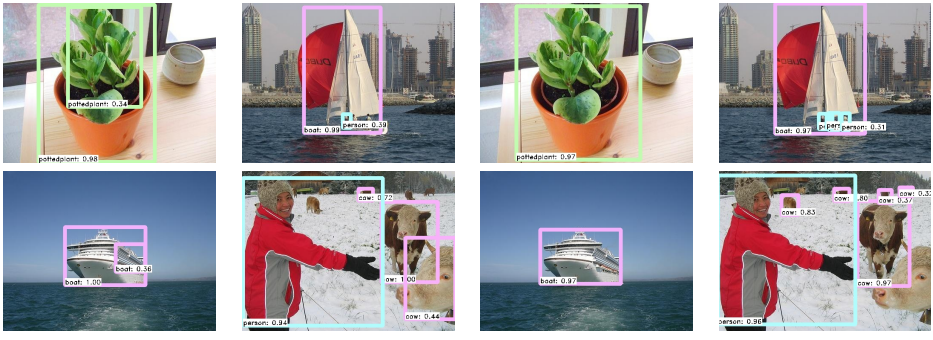
However, SSD does not consider the relationships between the different scales because it looks at only one layer for each scale. For example, in Fig.2(a), SSD finds various scale boxes for one object.

Second, SSD has the limitation that small objects are not detected well. This is not the problem only for SSD but the problem for most object detection algorithms. To solve this problem, there have been various attempts such as replacing the base network with more powerful one, e.g., replacing VGGNet with ResNet [4, 8] or increasing the number of channels in a layer [7]. Fig.2(b) shows that SSD has a limitation in detecting small objects. Especially, in the two figures, persons on the boat and small cows are not detected, respectively.

In this paper, we tackle these problems as follows. First, the classifier network is implemented considering the relationship between layers in the feature pyramid. Second, the number of channels (or feature maps) in a layer is increased efficiently. More specifically, only the layers in the feature pyramid are allowed to have increased number of feature maps instead of increasing the number of layers in the base network. The proposed network is suitable for sharing weights in the classifier networks for different scales, resulting in a single classifier network. This enables faster training speed with advanced generalization performance. Furthermore, this property of single classifier network is very useful in a small database. In the conventional SSD, if there is no training object at a certain size, the classifier network of that size cannot learn anything. However, if a single classifier network is used, it can get information about the object from the training examples in different scales.

Using the proposed architecture, our version of SSD can prevent detecting multiple boxes for one object as shown in Fig.2(c). In addition, the number of channels can be efficiently increased to detect small objects as shown in Fig.2(d). The proposed method shows state-of-the-art mAP (mean average precision) with a slightly degraded speed compared to the conventional SSD.

The paper is organized as follows. In Section 2, we briefly review the related works in the area of object detection, and then propose a different version of SSD, which is called as the *rainbow SSD*, in Section 3. The experimental results of the proposed algorithm are presented in Section 4. In Section 5, a discussion with some exemplary results of the proposed method is made. Finally, the paper is concluded in Section 6.



(a) (b) (c) (d)

Figure 2: Conventional SSD vs. the proposed Rainbow SSD (R-SSD). Boxes with objectness score of 0.3 or higher is drawn: (a) SSD with two boxes for one object; (b) SSD for small objects; (c) R-SSD with one box for one object; (d) R-SSD for small objects

2 Related Works

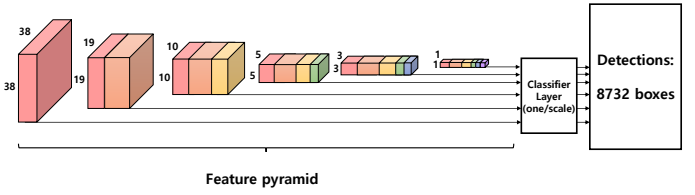
A wide variety of methods using deep learning have been applied to the problem of object detection and it continues to show performance improvements. In the earlier works pioneered by R-CNN (region-based CNN) [4], the candidate region was proposed through a separate algorithms such as selective search [16] or Edge boxes [19] and the classification was performed with deep learning.

Although R-CNN improved the accuracy using deep learning, speed was still a problem, and end-to-end learning was impossible. The region proposal network (RPN) was first proposed in faster R-CNN, which improved the speed of the object detector significantly and was able to learn end-to-end [15].

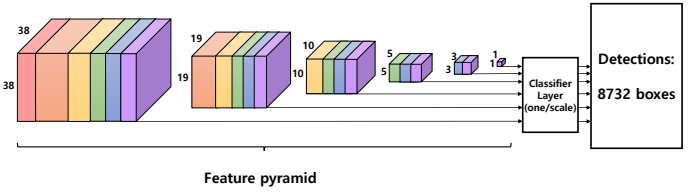
YOLO (you only look once) greatly improved the speed by dividing a single image into multiple grids and simultaneously performing localization and classification in each grid [14]. While YOLO performed object detection by concentrating only on speed, an enhanced version of YOLO, which is denoted as YOLO2, removed the fully connected layers and used anchor boxes to improve both the speed and the accuracy [13].

On the other hand, SSD creates bounding box candidates at a given position and scale and obtains their actual bounding box and score for each class [11]. Recently, to improve the accuracy of SSD, especially for small object, DSSD (deconvolutional SSD) that uses a large scale context for the feature pyramid was proposed [3]. DSSD applied a deconvolution module to the feature pyramid and used ResNet instead of VGGNet. DSSD succeeded in raising accuracy at the expense of speed.

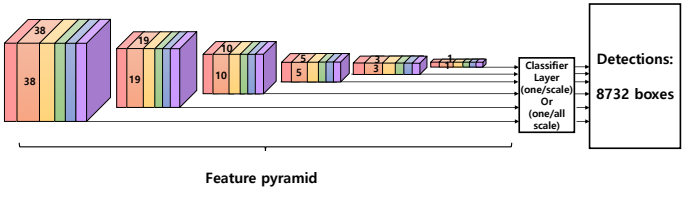
Besides the fields of object detection, the fields of segmentation have also been much developed by the application of deep neural networks. Many of them use pixel-based object classification in combination with upsampling or deconvolution to obtain the segmentation results in the same size as the input image [17, 18]. In addition, features are engineered in various ways by concatenation, element-wise addition or element-wise product with bypass, to obtain improved performance. This paper is inspired by the fact that we get improved abstract representation of the original image from features in different scales [1, 5].



(a) pooling



(b) deconvolution



(c) both pooling and deconvolution (Rainbow concatenation)

Figure 3: Proposed methods of feature concatenation: (a) concatenation through pooling (b) concatenation through deconvolution; (c) rainbow concatenation through both pooling and concatenation. (best viewed in color. See Fig. 1 for comparison.)

3 Method

As mentioned above, our strategy of improving the accuracy of SSD is to let the classifier network fully utilize the relationship between the layers in the feature pyramid without changing the base network that is closely located to the input data. In addition, it also increases the number of channels in the feature pyramid efficiently.

Figure 3 shows several ways of increasing the number of feature maps in different layers for the classifier networks to utilize the relationship between layers in the feature pyramid. To enable this, in Fig. 3(a), feature maps in the lower layers are concatenated to those of the upper layers through pooling. In this way, the classifier networks with large receptive fields can have enriched representation power for object detection. On the other hand, Fig. 3(b) shows the method of concatenating the feature maps of the upper layers to the lower layer features through deconvolution or upsampling. Fig. 3(c) shows the feature map concatenation method that utilize both the lower layer pooling and the upper layer deconvolution.

One thing to note is that before concatenating feature maps, a normalization step is inevitable. This is because the feature values in different layers are quite different in scale. Here, batch normalization [6, 10] is applied for each filter before concatenation.

All of the above methods have the advantage that end-to-end learning is possible. More details about each are described below.

# of box positions	38×38	19×19	10×10	5×5	3×3	1×1	total boxes
conventional SSD	4	6	6	6	4	4	8732
R-SSD with one classifier (4 boxes)	4	4	4	4	4	4	7760
R-SSD with one classifier (6 boxes)	6	6	6	6	6	6	11640

Table 1: The number of boxes for each classifier network and the number of total boxes

3.1 Concatenation through pooling or deconvolution

In the structure of SSD, generally, the numbers of channels in the lower layers are larger than those in the upper layer. To make explicit relationship between feature pyramid and to increase the number of channels effectively, we concatenate feature maps of the upper layers through pooling or concatenate feature maps of the lower layers through deconvolution. Unlike DSSD [3], which uses deconvolution module consisting of 3 convolution layer, 1 deconvolution layer, 3 batch normalization layer, 2 Relu and elementwise product, our model of concatenation through deconvolution performs only deconvolution with batch normalization and does not need elementwise product.

The advantage of these structure is that object detection can be performed with information from the other layers. On the other hand, the disadvantage is that information flows unidirectional and the classifier network cannot utilize other directional information.

3.2 Rainbow concatenation

As shown in Fig. 3(c), in the rainbow concatenation, pooling and deconvolution are performed simultaneously to create feature maps with an explicit relationship between different layers. After pooling or deconvolving features in every layers to the same size, we concatenate them. Using this concatenated features, detection is performed considering all the cases where the size of the object is smaller or larger than the specific scale. That is, it can have additional information about the object larger than or smaller than the object. Therefore, it is expected that the object in a specific size is likely to be detected only in an appropriate layer in the feature pyramid as shown in Fig. 2(c).

In addition, the low-layer features that has been with limited representation power are enriched by the concatenation of higher-layer features, resulting in good representation power for small object detection as in DSSD [3] without much computational overhead.

By rainbow concatenation, each layer of feature pyramid contains 2,816 feature maps (concatenation of 512, 1024, 512, 256, 256, and 256 channels) in total. Because each layer in the feature pyramid now has the same number of feature maps, weights can be shared for different classifier networks in different layers. Each classifier network of the conventional SSD checks 4 or 6 default boxes and the proposed rainbow SSD can unify default boxes in different layers with weight sharing. As shown in Table 1, conventional SSD makes a 8,732 total boxes. On the other hand, in the shared classifier with 4 default boxes for each layer, the number becomes 7,760, likewise, for the shared classifier with 6 default boxes, it becomes 11,640 in total.

3.3 Increasing number of channels

It is known that the larger the number of channels, the better the performance becomes. To demonstrate how efficient our method is, we present a comparison model, for which we simply change the number of channel in the original base network as shown in Table 2. As

	conv4		conv7		conv8		conv9 - conv11	
	1 - 3	4	1	2	1	2	1	2
SSD	512		1024		256	512	128	256
I-SSD	512	2048	1024	2048	1024	2048	1024	2048

Table 2: Number of channels in conventional SSD and I-SSD

	Input	Train	Test	mAP	FPS
YOLO[14]	448	VOC2007+2012	2007	63.4	45
YOLOv2[13]	416	VOC2007+2012	2007	76.8	**67**
YOLOv2 544x544[13]	544	VOC2007+2012	2007	78.6	40
Faster R-CNN[15]		VOC2007+2012	2007	73.2	5
R-FCN (ResNet-101)[8]		VOC2007+2012	2007	**80.5**	5.9
SSD*[11]	300	VOC2007+2012	2007	77.7	**61.1**
DSSD (ResNet-101)[8]	321	VOC2007+2012	2007	**78.6**	9.5
ISSD*	300	VOC2007+2012	2007	78.1	26.9
ours (SSD pooling)*	300	VOC2007+2012	2007	77.1	48.3
ours (SSD deconvolution)*	300	VOC2007+2012	2007	77.3	39.9
ours (R-SSD)*	300	VOC2007+2012	2007	**78.5**	35.0
ours (R-SSD one classifier (4 boxes))*	300	VOC2007+2012	2007	76.2	35.4
ours (R-SSD one classifier (6 boxes))*	300	VOC2007+2012	2007	77.0	34.8
SSD*[11]	512	VOC2007+2012	2007	79.8	**25.2**
DSSD (ResNet-101)[8]	513	VOC2007+2012	2007	**81.5**	5.5
ours (R-SSD)*	512	VOC2007+2012	2007	80.8	16.6

Table 3: VOC2007+2012 training and VOC 2007 test result (* is tested by ourselves)

in SSD, reduced VGG-16 pre-trained model is used as the base network. The number of channels in each convolution layer are set to be 2 to 8 times larger than the original network. From now on, this comparison network will be denoted as I-SSD, which stands for increased-channel SSD.

4 Experiments

In order to evaluate the performance of the proposed algorithm, we tested various versions of SSD for PASCAL VOC2007 [7]. In addition, we tested other datasets such as VOC2012 dataset [9], and MS COCO test-dev dataset [9].

4.1 VOC2007

We trained our model with VOC2007 and VOC2012 'trainval' datasets. The SSDs with 300×300 input were trained with a batch size of 8, and the learning rate was reduced from 10^{-3} to 10^{-6} by 10^{-1}. With each learning rate, we trained 80K, 20K, 20K, and 20K iterations respectively. The 512×512 input models were performed with a batch size of 4 and the learning rate was equal to that for 300×300 models. In the case of speed, it is measured by using the forward path of the network with a batch size of 1. The experiments were done with cuDNN v5.1 using CAFFE time function. Therefore, if the detection time is measured from the pre-processing (resizing image and so on), it may take longer. The experimental results are shown in Table 3. In the table, the performances of YOLO [14], YOLOv2 [13], Faster R-CNN [15], R-FCN [8], and DSSD [8] were obtained from their homepage [2] or the respective paper. To see the performance of various feature augmentation methods, we performed

[2] YOLO and YOLOv2 : http://pjreddie.com/darknet

Method	Train	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
YOLO [29]	07++12	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
YOLOv2 544 [29]	07++12	73.4	86.3	82.0	74.8	59.2	51.8	79.8	76.5	90.6	52.1	78.2	58.5	89.3	82.5	83.4	81.3	49.1	77.2	62.4	83.8	68.7
SSD300* [29]	07++12	75.6	88.2	82.9	74.0	61.1	47.3	82.9	78.9	91.6	57.8	80.2	63.9	89.3	85.5	85.9	82.2	49.7	78.8	73.1	86.6	71.6
DSSD321 [8]	07++12	76.3	87.3	83.3	75.4	64.6	46.8	82.7	76.5	92.9	59.5	78.3	64.3	91.5	86.6	86.6	82.1	53.3	79.6	75.7	85.2	73.9
R-SSD300*	07++12	76.4	88.0	83.8	74.8	60.8	48.9	83.9	78.5	91.0	59.5	81.4	66.1	89.0	86.3	86.0	83.0	51.3	80.9	73.7	86.9	73.8

Table 4: Results on VOC2012 test dataset using the networks trained by VOC07++12 train dataset (* is tested by ourselves)

	Average Precision			Average Precision			Average Recal #Dets			Average Recall		
	IoU=0.5:0.95	IoU=0.5	IoU=0.75	S	M	L	1	10	100	S	M	L
YOLOv2 [29]	21.6	44.0	19.2	5.0	22.4	35.5	20.7	31.6	33.3	9.8	36.5	54.4
SSD300[29]	25.1	43.1	25.8	6.6	25.9	41.4	23.7	35.1	37.2	11.2	40.4	58.4
DSSD321 [8]	28.0	46.1	29.2	7.4	28.1	47.6	25.5	37.1	39.4	12.7	42.0	62.6
R-SSD300	26.6	45.9	27.3	8.3	28.6	39.7	24.1	36.0	38.5	14.0	42.0	56.8

Table 5: Results on MS COCO test-dev2015 using the networks trained by trainval35k train dataset

experiments using features concatenated through pooling (SSD pooling) and deconvolution (SSD deconvolution) as described in Section 3.1. We also tested ISSD described in Section 3.3 for comparison. Three types of R-SSD was tested. The first one utilizes separate classifier networks for different scales and the rest two use a common classifier with 4 or 6 default boxes for a scale as described in Section 3.2. The conventional SSD was also trained and tested by ourselves.

ISSD: For the 300 input model, we experimented ISSD by increasing the number of the channels from 2 to 8 times for different layers as Table 2. As a result, there is a 0.4% mAP improvement in accuracy with 78.1% mAP compared to conventional SSD. However, due to the increased number of channels, the speed drops to 26.9 FPS

Concatenation through pooling or deconvolution: For the 300 input model, the concatenation through pooling and concatenation through deconvolution result in mAP of 77.1% and 77.3% respectively which is a degraded performance than that of conventional SSD by 0.6% and 0.4%. In addition, due to the increased complexity, the speed was also degraded to 48.3 FPS and 39.9 FPS, respectively.

R-SSD: For the 300 input model, there is a 0.8% improvement in accuracy with 78.5% mAP compared to conventional SSD. However, due to the increased computational complexity, the speed drops to 35.0 FPS. For the 512 input model, it results in mAP of 80.8% which is 1% better than conventional SSD. However its speed drops to 16.6 FPS. In particular, comparing the two SSD 512 models, precision increases 2.9% at recall value 0.8 and 8.2% at recall of 0.9. In the case of single classifier model with 300 input model, it has a 76.2% and 77.0% mAP when they use four and six default boxes, respectively.

4.2 VOC2012

Table 4 shows the results of VOC2012. All experiments were trained with VOC07++12 train dataset. YOLO and YOLOv2 have mAPs of 57.9% and 73.4%, respectively. SSD300 and DSSD321 show higher mAPs of 75.6% and 76.3%, respectively. Our R-SSD shows 76.4% mAP, which is higher than those of SSD300 and DSSD321. R-SSD is also faster than DSSD321.

4.3 MS COCO

Table 5 shows the results of MS COCO test-dev 2015. All experiments were trained with trainval35k train dataset. YOLOv2, SSD300 and DSSD321 have mAPs of 21.6, 25.1% and

tbp]

	Input	pre-trained model	mAP
SSD [11]	300	reduced VGG-16	66.1
ours (R-SSD)	300	reduced VGG-16	66.9
ours (R-SSD one classifier (6 boxes))	300	reduced VGG-16	67.2

Table 6: Results on VOC2007 test dataset trained with VOC2007 small train dataset (2,501 images)

	Input	Train	Recall						mAP@0.7+
			0.5	0.6	0.7	0.8	0.9	1	
SSD	300	VOC2007+2012	91.4	86.8	80.0	66.2	35.6	0	45.5
R-SSD (ours)	300	VOC2007+2012	92.7	88.4	82.4	68.9	37.6	0	47.2
SSD	512	VOC2007+2012	93.1	88.9	83.0	73.5	41.1	0	49.4
R-SSD (ours)	512	VOC2007+2012	92.8	89.8	84.9	76.4	49.3	0	52.7

Table 7: Average of precisions over all the classes for fixed recall and mAP@0.7+ (VOC evaluation Toolkit was used)

28.0% in IoU 0.5:0.95, respectively. Our R-SSD shows 26.6% mAP in IoU 0.5:0.95, which is higher than YOLOv2, SSD300 models.

4.4 Small Train Dataset in VOC2007

Table 6 is experimental results of different networks that were trained using the *train dataset* in VOC2007 which consists of a relatively small number of images (2,501 images in total). Each network was trained with these 2,501 images and the mAP was measured with VOC 2007 test dataset. The conventional SSD [11] shows a mAP of 66.1%, while R-SSD achieves 66.9% which is 0.8% better than that of SSD. Furthermore, R-SSD with one classifier achieves an even better mAP of 67.2%.

5 Discussion

5.1 New evaluation method

The most commonly used evaluation technique for object detection is mAP. AP (Average Precision) is a concept of integrating precision as recall is varied from 0 to 1 and mAP is defined as the average of AP for all the object classes.

In Table 7, we show the recall vs. average of precision table for PASCAL VOC 2007 test data. Note that average of precision here is averaged value for all the object classes. All of different versions of SSD (SSD 300, SSD 512, R-SSD 300, R-SSD 512) have almost the similarly high average of precision for small (< 0.5) recall values. Due to this, even if the precisions for large recall values show significant difference between algorithms (around 14% difference for SSD 300 and R-SSD 512), the difference in mAP is relatively small (around 3% for SSD 300 and R-SSD 512). Considering the use case of most object detection algorithms such as for autonomous vehicles, the precision value measured at high (> 0.7) recall is more important than that measured at small recall value. Therefore, in Table 7, we show the mAP computed by averaging only the APs at recall of 0.7 or higher. In the table, we can see more clearly the effectiveness of R-SSD over SSD. At recall of 0.9, R-SSD (512) outperformed SSD (512) by more than 8%. Note that the APs at recall of 1 is 0 for all the

cases. This is due to the fact that we cannot recall all the objects in the test images regardless how we lower the score threshold.

5.2 Concatenation by pooling or deconvolution

These two models are both inferior in accuracy and speed compared to conventional SSD, although they made explicit relationship between multiple layers and increased the number of channels. These two models need to perform more operations, therefore the speed can drop. As for accuracy, the reason can be conjectured that the layers sharing the same feature maps with other layers can be affected by the loss of other scales and do not fully focus on the scale. That is, they cannot learn properly on their scale.

5.3 Single classifier vs. Multiple classifiers

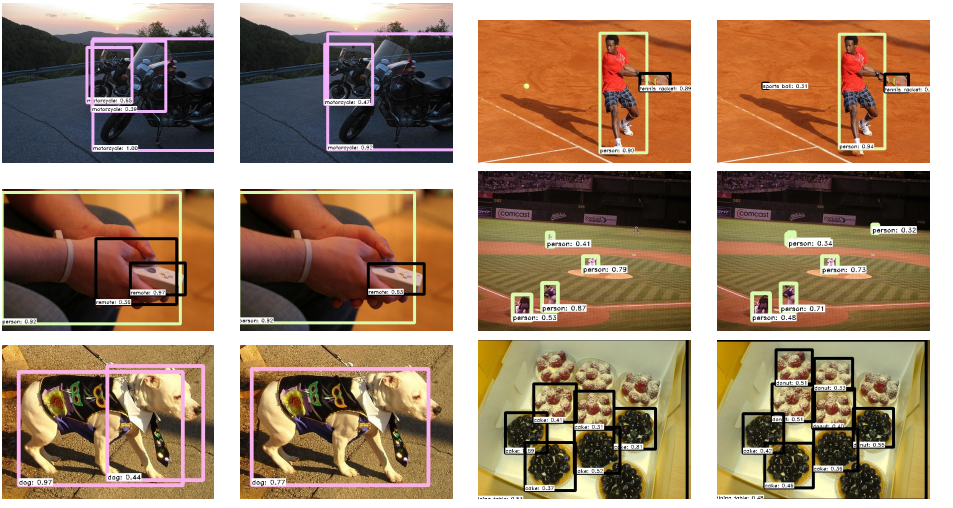
Unlike the conventional SSD, because R-SSD have the similar feature maps for different layers only different in size, the classifier network can be shared. Here, the experiments with a single classifier network by unifying the number of channels in each scale of feature pyramid. As shown in the Table 3, there is a difference in the number of boxes, but there is little difference in speed. In comparison, performance was 1.5 % and 0.7 % lower than that of conventional SSD. However, the advantage of a single classifier is that learning can be effective especially when there are significant imbalance between the numbers of training samples for different sizes. In this case, conventional SSD cannot train the classifier for a scale with small number of samples. However, in R-SSD, this problem is avoided because the classifier network is shared. Table 6 is results of VOC2007 test with small train dataset. It shows that training a single classifier is better not only in generalization power resulting in a faster training speed but also in detection performance when the training dataset is small. Furthermore, single classifier is faster at the early stage of training. Therefore, even for a large dataset, R-SSD can be trained fast by training a single classifier in the early stage and at a certain point, the classifiers can be trained separately for different scales.

5.4 Accuracy vs. Speed

The conventional SSD is one of the top object detection algorithms in both aspects of accuracy and speed. For SSD300 or SSD512, it has 77.7 % mAP and 79.8 % mAP respectively and has 61.1 FPS and 25.2 FPS. Looking at the results of the ISSD for comparison with our algorithm, ISSD had a 0.4 % accuracy gain, but the speed dropped to 26.9 FPS. In our experiments, R-SSD shows improved accuracy with a bit slow speed. Compared to the ISSD, R-SSD shows higher accuracy and faster speed. Moreover, it shows about 1% mAP improvement over the conventional SSD. At a speed of 15 fps or higher, our R-SSD shows 80.8 % mAP. Compared to R-FCN with similar accuracy, R-SSD is about three times faster.

5.5 Performances for different scales

Table 5 shows the results of MS COCO test-dev 2015 with average precision and average recall of each object size [9]. When the object size is small, R-SSD300 has higher average precision and average recall score than other methods. It can be shown that R-SSD misses a few small objects. At medium size, R-SSD300 also has higher or same average precision



(a) (b) (c) (d)

Figure 4: Conventional SSD vs. the proposed Rainbow SSD (R-SSD). Boxes with objectness score of 0.3 or higher is drawn: (a) SSD with two boxes for one object; (b) R-SSD with one box for one object (c) SSD for small objects; (d) R-SSD for small objects;

and average recall score. When the object size is large, however, R-SSD has lower average precision and average recall than even SSD300 model.

6 Conclusion

In this paper, we presented a rainbow concatenation scheme that can efficiently solve the problems of the conventional SSD. The contribution of the paper is as follows. First, it creates a relationship between each scale of feature pyramid to prevent unnecessary detection such as multiple boxes in different scales for one object. Second, by efficiently increasing the number of feature maps of each layer in the feature pyramid, the accuracy is improved without much time overhead. Finally, the number of feature maps for different layers are matched so that a single classifier can be used for different scales. By using a single classifier, improvement on the generalization performance can be expected, and it can be effectively used for datasets with size imbalance or for small datasets. The proposed R-SSD was created considering both the accuracy and the speed simultaneously, and shows state-of-the-art mAP among the ones that have speed of more than 15 FPS.

7 Acknowledgements

The research was supported by the Green Car development project through the Korean MTIE (10063267) and ICT R&D program of MSIP/IITP (2017-0-00306).

References

[1] Aayush Bansal, Xinlei Chen, Bryan Russell, Abhinav Gupta, and Deva Ramanan. Pixelnet: Towards a general pixel-level architecture. *arXiv preprint arXiv:1609.06694*, 2016.

[2] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

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

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

[5] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015.

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[7] Steve Lawrence, C Lee Giles, and Ah Chung Tsoi. What size neural network gives optimal generalization? convergence properties of backpropagation. Technical report, 1998.

[8] Yi Li, Kaiming He, Jian Sun, et al. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*, pages 379–387, 2016.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[10] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[12] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[13] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[14] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

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

[16] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104 (2):154–171, 2013.

[17] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2016.

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

[19] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.