Master's Thesis of Science in Engineering

Structural Improvements of Single Stage Networks for Object Detection

객체 검출을 위한
단일 단계 신경망의 구조적 개선

February 2018

Graduate School of
Convergence Science and Technology
Seoul National University
Department of Transdisciplinary Studies

Kyoungmin Lee

Structural Improvements of Single Stage Networks for Object Detection

Nojun Kwak

Submitting a master's thesis of Public Administration

February 2018

Graduate School of
Convergence Science and Technology
Seoul National University
Department of Transdisciplinary Studies

Kyoungmin Lee

Confirming the master's thesis written by
Kyoungmin Lee
February 2018

Chair	_____	(Seal)
Vice Chair	_____	(Seal)
Examiner	_____	(Seal)

Abstract

Recently, a lot of single stage detectors using multi−scale features have been actively proposed. They are much faster than two stage detectors that use region proposal networks (RPN) without much degradation in the detection performances. However, the feature maps in the lower layers close to the input which are responsible for detecting small objects in a single stage detector have a problem of insufficient representation power because they are too shallow. There is also a structural contradiction that the feature maps have to deliver low−level information to next layers as well as contain high−level abstraction for prediction.

In this paper, we propose a method to enrich the representation power of feature maps using Resblock and deconvolution layers. In addition, a unified prediction module is applied to generalize output results and boost earlier layers' representation power for prediction. The proposed method enables more precise prediction, which achieved higher score than SSD on PASCAL VOC and MS COCO. In

addition, it maintains the advantage of fast computation of a single stage detector, which requires much less computation than other detectors with similar performance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The development of deep neural networks (DNN) in recent years has achieved remarkable results not only in object detection but also in many other areas. In the early researches of object detection using DNN, much attention has been paid to representation learning that can replace handcrafted features without much consideration on the speed of detectors. Recently, real-time detectors with low computational complexities have been actively researched. Researches on two-stage detectors, mostly based on Faster R-CNN [21], applied the region proposal network (RPN) and RoI pooling to the feature maps extracted by a state-of-the-art classifier, such as ResNet-101 [10]. On the other hand, the single-stage methods such as YOLO [18] and SSD [17] removed RoI pooling layer and predict bounding boxes and corresponding class confidences directly while
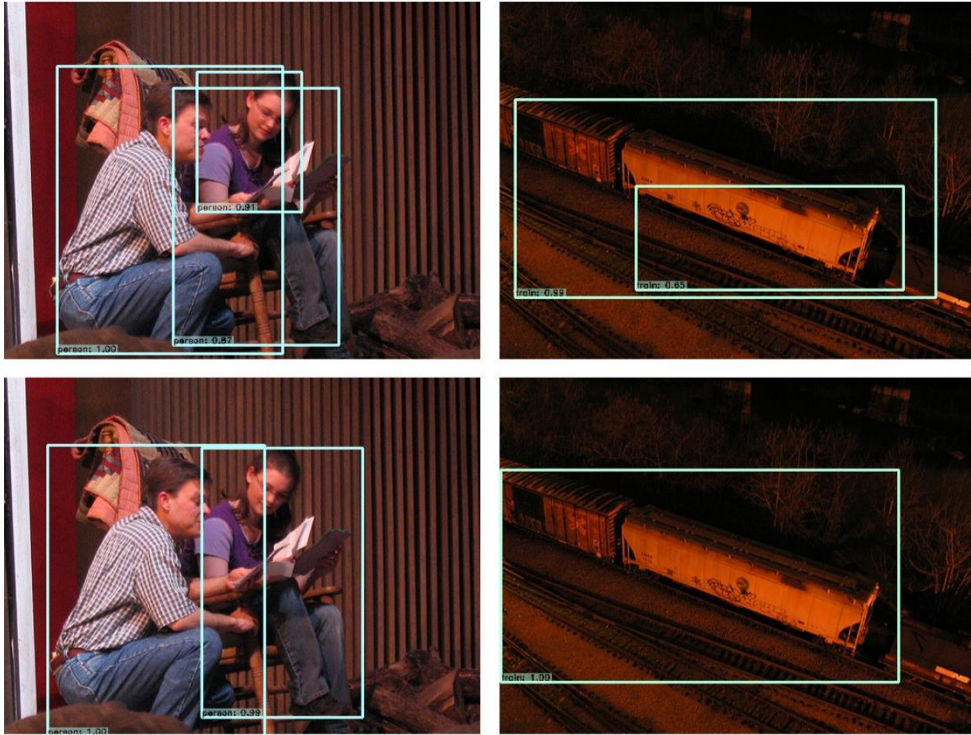
Figure 1: **Box-in-Box problem**. Top: SSD300. Bottom: RUN300
(proposed). SSD detects objects with overlapping boxes which are
redundant.

enabling faster detection and end-to-end learning. Especially SSD
makes use of multi-scale feature maps generated from a backbone
network such as VGG-16 [25] to detect objects in various sizes.
Since each of the prediction modules composed of 3×3 convolution
filters detects bounding box on each layer separately, they cannot
reflect appropriate contextual information from different scales. It
causes the problem named as "Box-in-Box" [11] as shown in Figure

1. In the figure, we can see that SSD often detects a single object with two overlapping boxes. The smaller box has partial image such as the upper body of a person or the head of an animal. To solve the problem, [6, 15] used ResNet and feature pyramid network (FPN) [14] structure to inject larger contextual information through deep convolutional back-bone by the use of deconvolution. However, these structures have the disadvantage of increasing the computational complexity, thus reduces detection speed, which is a key advantage of a single-stage detector. In this paper, we propose very simple ideas to solve the essential problems of multi-scale single stage detectors. First, we introduce a 3-way residual block, which is a structure where the Resblock [10] and the deconvolution layer are added on the multi-scale feature maps. It makes detected boxes be determined with larger context and be more reliable. Second, we integrate the multiple prediction modules, which had been applied separately to each layer, into one to boost information level of feature maps from earlier layers. The proposed structure, called "RUN; Residual features and Unified prediction Network", is a single-stage detector that combines 3-way Resblock with unified prediction module on VGG-16 network. RUN is not only very compact and fast compared to other ResNet-based two-stage detectors and single-stage detectors using FPN, but it also achieves superior or competitive performance compared to other competitors.

3

Chapter 2

Related Works

Overfeat[23], SPPNet[9], R-CNN[8], FastR-CNN[7], Faster R-CNN [21] and R-FCN [13] which are classified as region-based convolutional neural networks (R-CNN) showed a tremendous improvement in performance compared to the previous object detection techniques. These region-based approaches have achieved huge advances over the last few years and are still the state-of-the-art approaches among many object detection techniques. Specifically, these approaches usually use a two-stage method of generating a number of bounding boxes and then assigning a classification score to the bounding boxes. Thus, although classification may be relatively accurate, these are too slow to be used for real-time applications. Redmon et al. [18] proposed a

method named as YOLO to predict bounding boxes and associated class probabilities in a single step by framing object detection as a regression problem. It divides input images to grid maps and regresses bounding boxes for multiple objects on each grid. This was the beginning of single stage detection and subsequently inspired structures such as SSD [17]. However, since YOLO uses only the highest-level1 feature maps to detect objects, there is a lack of lower-level information, which results in somewhat inaccurate detection, especially for small objects. In order to solve this problem, SSD [17] utilized not only the highest-level features but also lower-level features which have enough resolution to detect small objects. As mentioned in Inside-Outside Net (ION) [1] and HyperNet [12], each feature maps at different layers have different abstraction levels for an input image. Therefore, it is clear that using multi-scale feature maps can improve detection performance for objects of various scales. In SSD, many default boxes are created in the feature maps and bounding box regression and classification are performed for each box area using 3× 3 convolution. This method enables multiscale object detection without using RoI pooling. In addition, it can effectively improve the detection accuracy of small objects which is a disadvantage of YOLO [18]. However, as mentioned in MS-CNN [2], SSD has the problem that back-propagation allows the gradient to cause unnecessary deformations in the feature maps since the

feature maps of the backbone network are used directly in bounding box regression and classification. Then, it can lead to some instability during learning. In addition, since each classifier only uses single scale feature maps, it cannot reflect larger or smaller contextual information other than the one for the corresponding scale. Recently, various methods have attempted to enhance the contextual information of each layer while taking advantage of SSD [17]. DSSD [6] could obtain higher accuracy by changing the base network to ResNet-101 [10] and combining the FPN [14] using deconvolution layers in combination with the existing multiple layers to reflect the large-scale context. However, with the use of deep structure of ResNet-101 and deconvolution layers, the processing speed degrades much (under 16.4 images per second), which prohibits the method to be used for real-time detection problems. Ren et al. [20] introduced a recurrent rolling convolution (RRC) architecture to improve detection performance by mutually complementing layers having different sizes of contextual information. RRC made multi-scale feature maps include large and small context by concatenating adjacent feature maps by pooling and deconvolution. This process was implemented by RNN structure and it allowed to reflect not only the information of the adjacent feature maps but also the information of the remote feature maps. Unlike RRC[20], Rainbow SSD(R-SSD)[11]proposed a method to concatenate feature maps not only in

the adjacent layers but also in all the layers for bounding box regression and classification using pooling and deconvolution. It achieves higher performance than SSD by enhancing representation power of feature maps. Also, by making the dimension of each layer the same, it made it possible to use a unified prediction module instead of different prediction modules for different layers. Woo et al. [27] proposed StairNet which utilizes both FPN [14] structure of base VGG-16 network and unified prediction of R-SSD. Additionally, Lin et al. [15] redefined the loss term for object detection which is named as FocalLoss. Unlike other batch reconstruction methods like OHEM [24], it effectively resolves the foreground-background imbalance problem by changing the loss term. Their RetinaNet which uses Focal Loss in combination with ResNet and FPN [14] structure achieved the state-of-the-art performance.

Chapter 3

Residual Feature and Unified Prediction Network for Single Stage Detection

 In this section, we propose residual feature maps and unified prediction module. It shows how the addition of a structurally simple idea can complement the drawbacks of SSD-based single-stage object detection methods.

3.1. Single-stage Detector

 Since Krizhevsky *et al.* made great achievements in ImageNet with CNN structure, many researches on structure using DNN have been
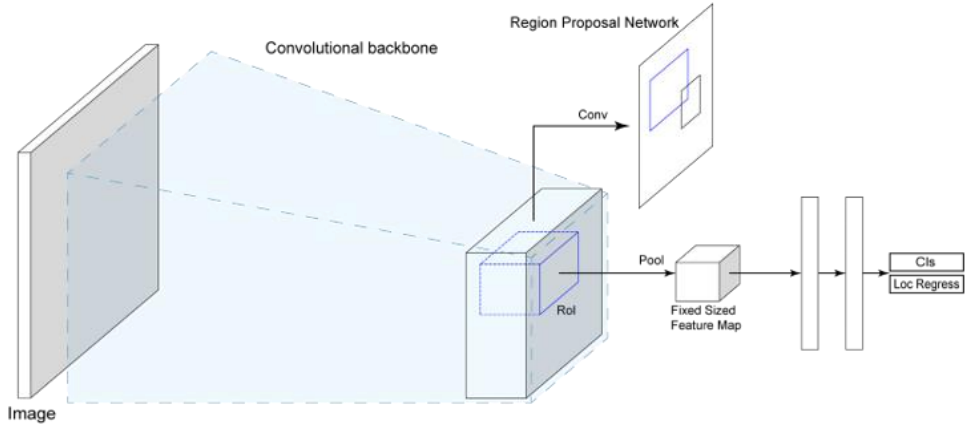
Figure 2: **Network of Faster R-CNN.** It shares feature map extracted by CNN with Region Proposal Network (RPN) and predictor.

carried out in various fields using image and it outperformed existing method, especially in object detection. R-CNN began the DNN-based object detection by extract CNN feature map from cropped images using selective search. This structure has evolved into Faster R-CNN [21] which are proposing a region in a feature map extracted by CNN and performing classification by pooling only features corresponding to the proposal box. These methods are called the two-stage detector.

The two-stage detector is a useful structure based on state-of-the-art research to date, but it has a disadvantage of speed that it needs to deploy for each RoI boxes. Wei *et al.* proposed a structure to improve these disadvantages in [17], which was a great inspiration
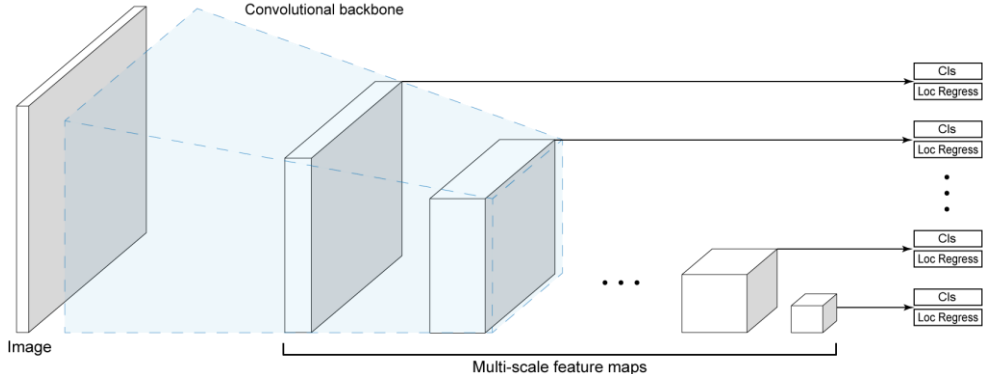
Figure 3: **Network of SSD**. It uses multi－scale feature maps and prediction modules.

for single stage detector studies.

The Single Stage Detector is aimed at eliminating the pooling of the Two Stage Detector and creating a structure for end－to－end learning and deployment. To do this, a multi－scale feature map is used as a prediction input, and a prediction is performed by applying an independent convolution layer to each feature map. At this time, the prediction module determined multi－class confidences. Then, the objective loss is a weighted sum of the localization loss and the confidence loss:

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \qquad (3.1)$$

where N is the number of matched proposal boxes. The confidence loss is the softmax loss over multiple classes confidences.

$$L_{conf}(x,c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} log(\hat{c}_i^p) - \sum_{i \in Neg} log(\hat{c}_i^0)$$

(3.2)

$$\text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

3.2. Residual Feature Maps

Recent CNN models designed for object detection makes use of a backbone network which is originally devised to solve image classification problems. Although the detection network can be trained end−to−end, the backbone network is normally initialized with the weights for the image classification problems. The relation between the features and predictions in the networks used for image classification can be expressed mathematically as follows:

$$x_n = \mathcal{F}_n(x_{n-1}) = (\mathcal{F}_n(x_{n-1}) \circ \mathcal{F}_{n-1} \circ \cdots \circ \mathcal{F}_1)(I) \qquad (3.3)$$

$$\textbf{Scores} = \mathcal{P}(x_n) \qquad (3.4)$$

where I is an input image, x_n is the nth−level feature map, $\mathcal{P}$ is a prediction function, and $\mathcal{F}_n$ is a combination of nonlinear transformations such as convolution, pooling, ReLU, etc. Here, the top feature map, x_n, learns information on high−level abstraction. On the other hand, $x_k(k < n)$ has more local and low−level information

1 1

as k becomes smaller.

SSD [17] applies several feature maps with different scales directly as an input to separate prediction modules to calculate object positions and classification scores, which can be denoted by the following equation:

$$\text{Detection} = \left\{ \mathcal{P}_1(x_{s_1}), \mathcal{P}_2(x_{s_2}), \cdots, \mathcal{P}_k(x_{s_k}) \right\} \qquad (3.5)$$

where s_1 to s_k are feature indices for source feature maps for multi−scale prediction, $\mathcal{P}_k$ is a function that outputs multiple objects with different positions and scores. Combining (1) and (3), it can be expressed as

$$\text{Detection} = \left\{ \mathcal{P}_1(x_{s_1}), \mathcal{P}_2\left(\mathcal{F}_{s_1}^{s_2}(x_{s_1}) \right), \cdots, \mathcal{P}_k\left(\mathcal{F}_{s_1}^{s_k}(x_{s_1}) \right) \right\} \qquad (3.6)$$

Where $\mathcal{F}_a^b(x_a) \triangleq (\mathcal{F}_b \circ \cdots \circ \mathcal{F}_{a+1})(x_a)$. Here, the earlier feature map x_{s_1} needs to learn high−level abstraction to improve the performance of $\mathcal{P}_1(x_{s_1})$. At the same time, it also needs to learn local features for efficient information transfer to the next feature maps. This not only makes learning difficult, but also causes the overall performance to decrease.
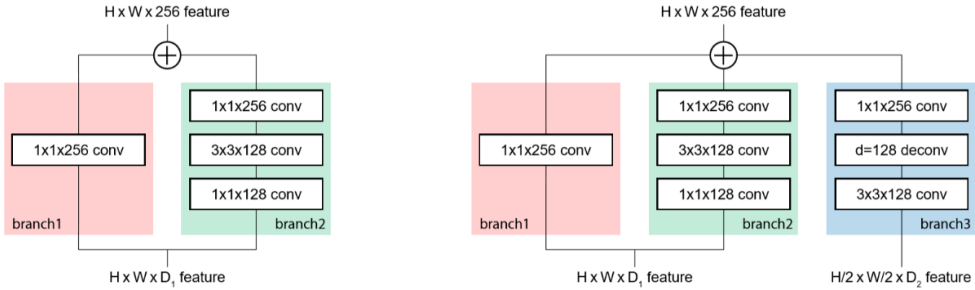
Figure 4: **Residual blocks**. Left: 2-way Resblock. Right: 3-way Resblock with deconvolution branch (branch3).

 To resolve this problem, SSD [17] added L2 normalization layer between the conv4_3 layer and the prediction module, which results in a reduced magnitude of the gradients from the prediction module. Cai *et al*. [2] tried to solve this problem by adding a convolution layer only to the conv4_3 layer. Since the above problem is not solely on the conv4_3 layer, the aforementioned approaches do not essentially solve the problem. To meet this contradictory requirement of maintaining low-level information while having the flexibility to learn high-level abstraction, it is desired to separate and decouple the backbone network and the prediction module in the training phase.

 In order to solve the same problem, we propose a new architecture that decouples backbone network from the prediction module. Instead of directly connecting the feature maps in the backbone network to the prediction module, we inserted a multi-way Resblock for each level of feature maps, which acts like a bumper. The detailed
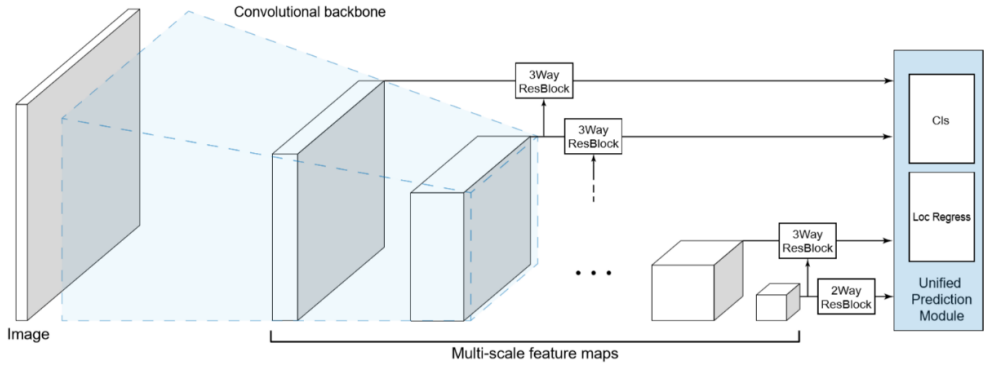
Figure 5: **Networks of RUN**. Compared with SSD, RUN has residual blocks and unified prediction module. The arrow from the bottom to the top indicates the deconvolution branch

architecture of the proposed multi-way Resblocks are shown in Figure 4. Convolution layers and nonlinear activation units are used for all branches of the proposed Resblock. This prevents the gradients of the prediction module from flowing directly into the feature maps of the backbone network. Also, it clearly distinguishes the features to be used for prediction from the features to be delivered to the next layer. In other words, the proposed Resblock takes the role of learning high−level abstraction for object detection, while the backbone network containing low−level features is designed to be intact from the high−level detection information. This design helps to improve the feature structure of the SSD [17] by forcing it not to learn highlevel abstraction and to keep low−level image features.

1 4

Also, the depths of the earlier layers (eg. conv4_3) used for small−sized object detection in SSD [17] are very shallow. Therefore, in SSD, small objects cannot be detected well because the representation power is insufficient to be used in the prediction as it is. To supplement this problem, we used a 3 × 3 convolution layer in branch2 of the Resblock as shown in Figure 5 to reflect the peripheral contextual information.

Branch3 in the right side of Figure 5 contains a deconvolution layer whose input is the feature maps of the consecutive layer. This is similar to a structure proposed in [6] and [20], and it is a proper method to propagate large contextual information to a small−scale feature map so that even when detecting a small object, information about its surroundings is also utilized. This can reduce the cases of detecting a part of an actual object. Thus, it can be a remedy for the box−in−box problem described earlier. Finally, the proposed architecture in Figure 5 can be expressed as follows:

$$
\begin{aligned}
\textbf{Detection} \\
= \left\{ \mathcal{P}_1\left(\widehat{x}_{s_1,s_2}\right), \mathcal{P}_2\left(\widehat{x}_{s_2,s_3}\right), \cdots, \mathcal{P}_{k-1}\left(\widehat{x}_{s_{k-1},s_k}\right), \mathcal{P}_k\left(\widehat{x}_{s_k}\right) \right\}
\end{aligned}
\tag{3.7}
$$

where $\widehat{x}_{a,b} = \mathcal{B}_1(x_a) + \mathcal{B}_2(x_a) + \mathcal{B}_3(x_b)$ and $\widehat{x}_a = \mathcal{B}_1(x_a) + \mathcal{B}_2(x_a)$. Here, $\mathcal{B}_1$, $\mathcal{B}_2$ and $\mathcal{B}_3$ indicate branch1, branch2 and branch3, respectively.

3.2. Unified Prediction Module

Detecting objects of various sizes has been recognized as an important problem in object detection. Traditionally, [26, 3, 4] used a single classifier to predict multi-scale feature maps extracted from the image pyramid. There is another approach of using multiple classifiers on a single input image. The latter has the advantage of reducing the amount of computation for calculating feature maps. However, it requires an individual classifier for each object scale.

Since the neural network has been prevalent, the two-stage detectors applied RoI Pooling to the CNN output to extract feature maps of the same size from objects of different sizes. These feature maps were used as the input of a single classifier. Meanwhile, other methods using multiscale features, such as SSD [17], adopted multiple classifiers since feature maps in each scale differed not only in length but also in the underlying contextual information. In order to effectively learn the prediction layers of various scales, it is necessary to input objects of various scales. SSD could dramatically increase the detection performance through augmentation which transforms the size of input images.
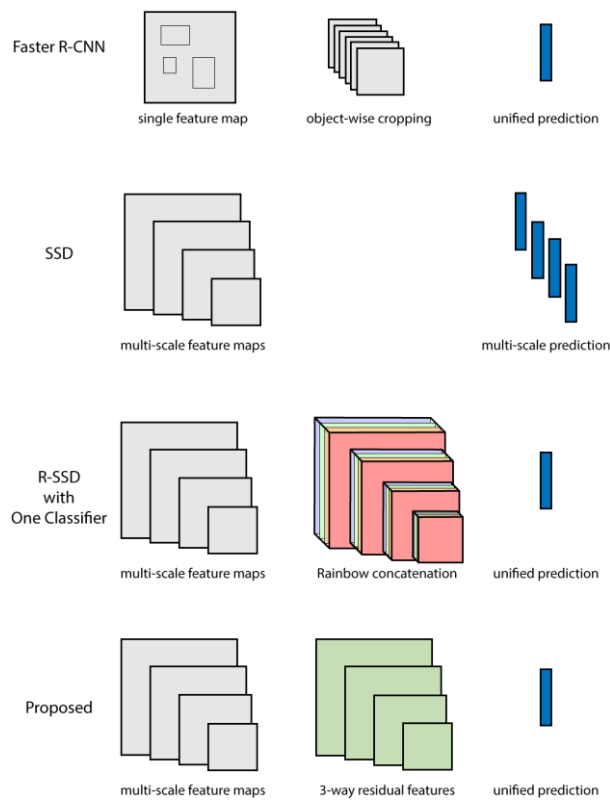
Figure 6: **Comparison of various object detection schemes**: a) R−CNN and its variants need object−wise cropping and the prediction is done by a common unified classifier. b) SSD does not need any cropping but requires a separate classifier for each scale of feature maps. c) R−SSD concatenates feature maps in different layers so that objects in each scale can be predicted with one unified classifier with the same amount of information. d) In the proposed method, Resblock takes the role of feature map concatenation and one unified classifier is used for prediction.

R-SSD [11] proposed the Rainbow concatenation which combines feature maps in different scales using pooling and deconvolution. This allows to set the depth of the input features for each prediction module to be the same. Thus, R-SSD could use a single classifier that shares the weight of multi-scale prediction modules. Similarly, the proposed 3-way Resblocks enforce all the feature maps to have the same depth of 256 as shown in Figure 6. Thus, structurally, it is possible to unify convolution layers of different prediction modules like R-SSD. The idea of the unified prediction module is similar to [11], but our method is different from R-SSD in information contained in the input feature maps.

This approach makes differently-scaled feature maps have similar level of information. SSD [17] used multiple features of various scales. This results in an improved performance of detecting small objects compared with YOLO [18, 19] which used only the last layer of the back-bone network. However, since its earliest feature map is obtained from much shallower layers than the later feature maps, it still has a limitation of insufficient information for prediction. Because unified prediction applies equally to feature maps of all scales, it forces the output of the 3-way Resblock between the feature map of the backbone and This approach makes differently-scaled feature maps have similar level of information. SSD [17] used multiple features of various scales. This results in an improved

performance of detecting small objects compared with YOLO [18, 19] which used only the last layer of the back−bone network. However, since its earliest feature map is obtained from much shallower layers than the later feature maps, it still has a limitation of insufficient information for prediction. Because unified prediction applies equally to feature maps of all scales, it forces the output of the 3−way Resblock between the feature map of the backbone and the prediction module to be learned at a similar information level. It means that unified prediction in combination with the residual feature block makes the feature maps in the earliest Resblocks rich in context. A brief summary of different object detection schemes is shown in Figure 6.

Chapter 4

Experiment

We experimented the proposed method on PASCAL VOC 2007 [5], PASCAL VOC 2012 and MS COCO datasets [16]. Our implementation is based on the publicly available SSD [17][①]. All of experiments results of SSD are the latest scores with data augmentation mentioned in [6]. For all the experiments, the reduced VGG-16 model [25] pre-trained on the ILSVRC CLS-LOC dataset [22] is used as the backbone network. For fair comparison, most of the settings are set to be the same as those of SSD except the number of proposals. It is different from SSD, because we used 6 default boxes in all the prediction layers for unified prediction while SSD used 4 for the conv4 3 and the top layer, and 6 for the rest.

---

① https://github.com/weiliu89/caffe/tree/ssd

Ablation Study on PASCAL VOC2007 We trained our model on VOC2007 trainval and VOC2012 trainval. We set the batch size as 32. For the training of the 2−way model, we used learning rate of 10^{-3} initially, then it decreased by a factor of 10 at 80k and 100k iterations, respectively. The training was terminated at 120k iterations. For the 3−way model, we froze all the weights of the pre−trained 2−way model except the prediction module, then fine−tuned the network using the learning rate of 10^{-3} for 40k iterations, 10^{-4} for the next 20k iterations, and 10^{-5} for the final 10k iterations. The end−to−end training was also applied on the 3−way model, but the results were worse than the above training strategy.

Method	mAP
SSD 300	77.5
SSD 300 + 2WAY	78.3
SSD 300 + 2WAY + Unified Pred	78.6
SSD 300 + 3WAY	78.8
SSD 300 + 3WAY + Unified Pred	**79.2**

Table 1: **PASCAL2007** test detection results.

Table 1 shows our result on PASCAL VOC 2007 test set. Here, Unified Pred is the proposed unified prediction module and the prediction modules for the ones without this indication were trained separately as in the original SSD. As mentioned above, each 3−way model was fine−tuned on the corresponding 2−way model. In this

Method	data	network	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SSD300 [17]	07++12	VGG	75.8	88.1	82.9	74.4	61.9	47.6	82.7	78.8	91.5	58.1	80.0	64.1	89.4	85.7	85.5	82.6	50.2	79.8	73.6	86.6	72.1
SSD321 [6]	07++12	Residual-101	75.4	87.9	82.9	73.7	61.5	45.3	81.4	75.6	92.6	57.4	78.3	65.0	90.8	**86.8**	85.8	81.5	50.3	78.1	75.3	85.2	72.5
DSSD321 [6]	07++12	Residual-101	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 [11]	07++12	VGG	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
StairNet [27]	07++12	VGG	76.4	87.7	83.1	74.6	64.2	51.3	83.6	78.0	92.0	58.9	81.8	**66.2**	89.6	86.0	84.9	82.6	50.9	80.5	71.8	86.2	73.5
RUN2WAY300	07++12	VGG	76.2	**88.4**	83.2	73.7	63.0	50.2	82.6	79.2	91.3	58.6	81.4	64.8	90.0	85.9	85.4	82.8	50.5	81.3	74.0	86.1	72.4
RUN3WAY300	07++12	VGG	**77.1**	88.2	**84.4**	**76.2**	63.8	**53.1**	82.9	**79.5**	90.9	**60.7**	82.5	64.1	89.6	86.5	**86.6**	**83.3**	51.5	**83.0**	74.0	**87.6**	**74.4**

Table 3: **SSD300−based models on PASCAL 2012 test.**

Trained with 07++12 (07trainval+07test+12trainval).

Method	data	network	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
ION [1]	07+12+S	VGG	76.4	87.5	84.7	76.8	63.8	58.3	82.6	79.0	90.9	57.8	82.0	64.7	88.9	86.5	84.7	82.3	51.4	78.2	69.2	85.2	73.5
Faster [10]	07++12	Residual-101	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
R-FCN [13]	07++12	Residual-101	77.6	86.9	83.4	**81.5**	63.8	**62.4**	81.6	81.1	93.1	58.0	83.8	60.8	92.7	86.0	84.6	84.4	**59.0**	80.8	68.6	86.1	72.9
SSD512 [17]	07++12	VGG	78.5	90.0	85.3	77.7	64.3	58.5	85.1	84.3	92.6	61.3	83.4	65.1	89.9	88.5	**88.2**	85.5	54.4	82.4	70.7	87.1	75.6
SSD 513 [6]	07++12	Residual-101	79.4	90.7	**87.3**	78.3	66.3	56.5	84.1	83.7	94.2	62.9	84.5	66.3	92.9	88.6	87.9	85.7	55.1	83.6	74.3	**88.2**	76.8
DSSD 513 [6]	07++12	Residual-101	**80.0**	**92.1**	86.6	80.3	**68.7**	58.2	84.3	85.0	**94.6**	63.3	**85.9**	65.6	**93.0**	88.5	87.8	86.4	57.4	**85.2**	73.4	87.8	**76.8**
RUN2WAY512	07++12	VGG	79.3	89.7	87.1	79.2	65.6	61.3	**85.3**	85.0	92.9	60.6	83.8	**66.4**	90.6	88.6	88.1	86.1	54.8	84.6	72.5	87.4	75.8
RUN3WAY512	07++12	VGG	79.8	90.0	**87.3**	80.2	67.4	**62.4**	84.9	**85.6**	92.9	61.8	84.9	66.2	90.9	**89.1**	88.0	**86.5**	55.4	85.0	72.6	87.7	**76.8**

Table 2: **SSD500−based models and other two−stage detectors on PASCAL 2012 test**. Trained with 07++12.

experiment, we observed that the proposed model with only 2−way Resblock without the deconvolution path achieved 1.1% higher mAP than that of SSD. The 3−way model which further utilizes deconvolution layers was up to 0.6% higher than the 2−way model. The unified prediction module made better advance in the 3−way model than the 2−way model, which scored 79.2% and 78.4% respectively.

 PASCAL VOC 2012 For VOC 2012 test, we trained models on 07++12 dataset consisting 07trainval, 07test and 12trainval. First, we performed an experiment applying the 2−way Resblock in

combination with the unified prediction, then, another experiment was performed using the 3－way Resblock with unified prediction after freezing the weights of the contained 2－way Resblock.

Table 3 shows the VOC 2012 test results of RUN300 and other

Method	data	network	Avg. Precision, IoU:			Avg. Precision, Area:			Avg. Recall, #Dets:			Avg. Recall, Area:		
			0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
Faster [21]	trainval	VGG	21.9	42.7	-	-	-	-	-	-	-	-	-	-
ION [1]	train	VGG	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
R-FCN [13]	trainval	Residual-101	29.9	51.9	-	10.8	32.8	45.0	-	-	-	-	-	-
RetinaNet [15]	trainval	Residual-101	**39.1**	**59.1**	**42.3**	**21.8**	**42.7**	**50.2**	-	-	-	-	-	-
SSD300 [17]	trainval35k	VGG	25.1	43.1	25.8	6.6	25.9	41.4	23.7	35.1	37.2	11.2	40.4	58.4
SSD321 [6]	trainval35k	Residual-101	28.0	45.4	29.3	6.2	28.3	49.3	25.9	37.8	39.9	11.5	43.3	64.9
DSSD321 [6]	trainval35k	Residual-101	28.0	46.1	29.2	7.4	28.1	47.6	25.5	37.1	39.4	12.7	42.0	62.6
RUN2WAY300	trainval35k	VGG	27.4	46.1	28.4	8.9	27.9	43.8	25.0	37.3	39.5	14.6	42.6	59.8
RUN3WAY300	trainval35k	VGG	28.0	47.5	28.9	9.9	28.6	43.9	25.3	38.0	40.5	16.2	43.8	60.2
SSD512 [17]	trainval35k	VGG	28.8	48.5	30.3	10.9	31.8	43.5	26.1	39.5	42.0	16.5	46.6	60.8
SSD513 [6]	trainval35k	Residual-101	31.2	50.4	33.3	10.2	34.5	49.8	28.3	42.1	44.4	17.6	**49.2**	65.8
DSSD513 [6]	trainval35k	Residual-101	33.2	53.3	35.2	13.0	35.4	51.1	**28.9**	**43.5**	**46.2**	21.8	49.1	**66.4**
RUN2WAY512	trainval35k	VGG	31.7	52.1	33.6	13.2	33.9	46.5	27.7	42.2	44.7	22.0	47.9	62.7

Table 4: **MSCOCO test－dev detection results.**

models based on SSD300 [17]. The proposed model, RUN300, has a big performance improvement compared to the base model SSD300. Especially, the 3－way model achieved 77.1% mAP, outperforming other SSDbased models. In addition, it showed improvement of 0.7% mAP compared to StairNet [27] which uses FPN [14] and unified prediction. From this result, we can conjecture that the proposed 3－way Resblock is more effective than FPN.

Table 2 shows results of RUN512 models and others. The 3－way model achieved 79.8% mAPs, which is 1.3% better than that of SSD512 [17]. It performs slightly worse than DSSD513 [6], which is probably because the ResNet101 [10] backbone of DSSD513

produces better features for larger input images than VGG–16 [25] of SSD and ours.

COCO For fair comparison with SSD [17], most of the hyper-parameters required for training were set to the same as SSD. For training 2–way models, we used a learning rate of 10^{-3} for the first 240k iterations, 10^{-4} for the next 120k iterations and 10^{-5} for the last 40k. For training 3–way models, we used a learning rate of 10^{-3} for the first 120k iterations, 10^{-4} for the next 60k iterations and 10^{-5} for the last 20k, which are exactly half of those for the 2–way models. Other parameters such as scales and aspect ratios of the prior box were identical to those of SSD.

Table 4 shows the performance of various methods on MS COCO test–dev. Despite the proposed methods use a relatively shallow network, VGG–16 [25], they achieved enough performance to compare with other methods which use a very deep network. The fourth column indicates that RUN3WAY300 achieved 2.9% better mAP compared to SSD300 [17]. It was the same performance with SSD321 and DSSD321 [6], which adopted ResNet–101 [10] as their back–bone network. Also, RUN3WAY512 achieved 3.6% better mAP than SSD512. In particular, RUN3WAY512 achieved the highest average precision and recall for small objects among compared methods except RetinaNet. It means that the proposed Resblock is a quite effective module to enhance low–level feature maps.

Method	network	mAP	FPS	GPU
Faster R-CNN [21]	VGG16	73.2	7	Titan X
Faster R-CNN [10]	Residual-101	76.4	2.4	K40
R-FCN [13]	Residual-101	80.5	9	Titan X
SSD300 [17]	VGG16	77.5	54.5*	Titan X
SSD321 [6]	Residual-101	77.1	16.4	Titan X
DSSD321 [6]	Residual-101	78.6	11.8	Titan X
R-SSD300 [11]	VGG16	78.5	37.1*	Titan X
StairNet [27]	VGG16	78.8	30	Titan X Pascal
RUN2WAY300	VGG16	78.6	41.8	Titan X
			58.4	Titan X Pascal
RUN3WAY300	VGG16	79.2	40.0	Titan X
			56.3	Titan X Pascal
SSD512 [17]	VGG16	79.5	24.5*	Titan X
SSD513 [6]	Residual-101	80.6	8.0	Titan X
DSSD513 [6]	Residual-101	81.5	6.4	Titan X
R-SSD512 [11]	VGG16	80.8	15.8*	Titan X
RUN2WAY512	VGG16	80.6	20.1	Titan X
			31.8	Titan X Pascal
RUN3WAY512	VGG16	80.9	19.5	Titan X
			29.8	Titan X Pascal

Table 5: **Speed & Accuracy on PASCAL VOC2007 test**. ∗ is measured by ourselves.

Speed vs Accuracy The single stage detectors, which are represented by YOLO [18] and SSD [17], proposed endto-end neural networks that removed the RoI Pooling of two-stage detectors. They have achieved a lot of speed improvements, but they could not avoid the loss of accuracy. Conversely, recent single stage detectors have been studied to improve performance, while suffering the loss of speed. Unlike other approaches, the proposed RUN is designed to maximize performance at high speeds on the VGG-16 [25] backbone, which has significantly fewer layers and parameters

than ResNet [10]. The experimented results demonstrate the performance improvement of RUN.
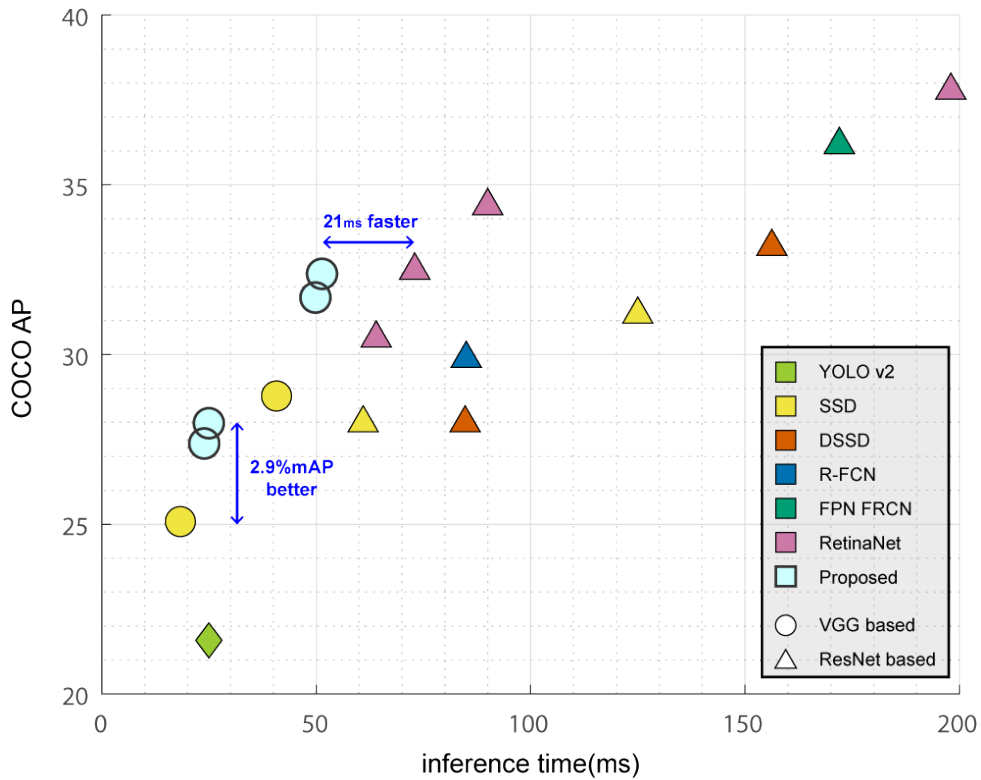

Figure 7: **Speed vs. Accuracy of recent methods using public numbers on COCO.** Our results (sky blue circles) are measured on Titan X. (Best viewed in color.)

Table 5 shows that our method outperforms other competitors with less loss of speed. Our experiments were tested using Titan X GPU, cuDNN v5.1 and Intel I76700@3.4GHz. For exact comparison, we measured FPS of some methods on the same environment and marked ∗ in the table.

In Figure 7, we show the trade-off relation between the detection accuracy and inference time by plotting the results of RUN and other methods on COCO test-dev. The RUN-3way-300 model (25.0ms, 28.0% mAP) is 36% slower but 2.9% better in mAP than the SSD300 [17] model (18.3ms, 25.1% mAP). It is about 60% faster than ResNet101 based SSD321 [6] model (61ms, 28.0% mAP) that has a similar performance. Likewise, the RUN-3way-512 (51.4ms, 32.4% mAP) is 26% slower but 3.6% better in mAP than the SSD512 model (40.8ms, 28.8% mAP). It is about 44% faster than RetinaNet-50-500 [15] (73ms, 32.5% mAP) that has a similar performance.

In addition, we measured FPS of our methods on Titan X Pascal with the other environment kept the same. Table 5 shows that even the most complex version of our method, RUN3WAY512, can works in real time (29.8 FPS) on Tital X Pascal.

Chapter 5

Conclusion

The proposed RUN architecture for object detection was originated from the awareness of the contradictory requirements for multi－scale features that they should contain low level information on an image as well as high－level information on objectness. The proposed 3－way Resblock alleviated the gradient exploitation problem and enriched contextual information, an important element of prediction. We also showed that the generalization performance of multiscale prediction can be improved by integrating the separate prediction modules into one unified prediction module. This approach, which can be seen to be somewhat simple, resulted in outstanding performance on the PASCAL VOC test. The results on COCO dataset also show how fast and efficient our algorithms are. We expect the proposed method be not restricted to SSD－based methods but also applicable

to other structures utilizing multi-scale features.

Figure 8: Detection examples of RUN300 3-way(Right) compared with SSD300(Left).

Bibliography

[1] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick. Insideoutside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2874–2883. IEEE, 2016.

[2] Z. Cai, Q. Fan, R. Feris, and N. Vasconcelos. A unified multiscale deep convolutional neural network for fast object detection. In *ECCV*, 2016.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[4] P. Dollar, R. Appel, S. Belongie, and P. Perona. Fast feature´ pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014.

[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A.

Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303– 338, 2010.

[6] C.−Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[7] R. Girshick. Fast r−cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440– 1448, 2015.

[8] R. Girshick, J. Donahue, T. Darrell, and J. 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.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[11] J. Jeong, H. Park, and N. Kwak. Enhancement of ssd by concatenating feature maps for object detection. *arXiv preprint arXiv:1705.09587*, 2017.

[12] T. Kong, A. Yao, Y. Chen, and F. Sun. Hypernet: Towards accurate region proposal generation and joint object detection.

In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 845–853. IEEE, 2016.

[13] Y. Li, K. He, J. Sun, et al. R–fcn: Object detection via regionbased fully convolutional networks. In *Advances in Neural Information Processing Systems*, pages 379–387, 2016.

[14] T.–Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and´ S. Belongie. Feature pyramid networks for object detection. *arXiv preprint arXiv:1612.03144*, 2016.

[15] T.–Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar.´ Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017.

[16] T.–Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Com–´ mon objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[18] J. Redmon, S. Divvala, R. Girshick, and A. 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.

[19] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[20] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu. Accurate single stage detector using recurrent rolling convolution. In *CVPR*, 2017.

[21] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[22] Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.

[23] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[24] Shrivastava, A. Gupta, and R. Girshick. Training regionbased object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–769, 2016.

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[26] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137– 154, 2004.

[27] S. Woo, S. Hwang, and I. S. Kweon. Stairnet: Top-down semantic aggregation for accurate one shot detection. *arXiv preprint arXiv:1709.05788*, 2017.

요 약

다중 스케일 특징 지도를 사용한 단일 단계 검출기는 최근 활발하게 등장하고 있다. 이러한 구조는 지역 제안 신경망을 사용하는 이중 단계 검출기에 비해 적은 성능 손실에서 훨씬 더 빠른 속도를 내고있다. 하지만, 작은 객체를 검출하기 위해 사용하는 입력과 가까운 낮은 단계의 층에서 추출한 특징 지도는 너무 얕기 때문에 충분한 표현력을 가질 수 없다는 한계점이 있다. 또한, 특징 지도가 다음 층에 전달하기 위한 저 수준의 정보 습득과 예측을 위한 고수준의 추상화를 동시에 달성해야 한다는 구조적 모순에 부딪힌다.

이 논문에서는 잔여 블록(Resblock)과 디컨볼루션 층을 사용하여 특징 지도의 표현력을 강화하는 방법을 제안한다. 추가적으로 출력 결과를 일반화함과 동시에 낮은 층의 표현력을 배가할 수 있는 통합된 추정 모듈을 적용하였다. 제안한 방법은 더 정확한 추정을 가능하도록 하여 PASCAL VOC와 MS COCO 데이터셋에서 SSD보다 높은 점수를 달성하였다. 게다가 단일 단계 검출기의 장점인 속도를 유지하여 비슷한 성능의 다른 검출기보다 훨씬 적은 연산량을 필요로 한다.

주요어 : 객체 검출, 인공 신경망, 딥러닝

학번 : 2016-26041