Residual Features and Unified Prediction Network for Single Stage Detection

Kyoungmin Lee  Jaeseok Choi  Jisoo Jeong  Nojun Kwak
Seoul National University
{strx2322, jaeseok.choi, soo3553, nojunk}@snu.ac.kr

Abstract

Recently, a lot of single stage detectors using multi-scale features have been actively proposed. They are much faster than two stage detectors using 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 used to detect 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 deconvolutional layers. In addition, a unified prediction module is applied to generalize output results. The proposed method enabled more precise prediction and scored 77.0 % mAP on PASCAL VOC 2012 test set, which is 1.2 % higher than that of SSD300. In addition, it takes 15.6 milliseconds for Titan X Pascal GPU, which indicates that it maintains the advantage of fast computation of a single stage detector.

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, 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. On the other hand, the single-stage methods such as YOLO [12] and SSD [11] removed RoI pooling layer and predict bounding boxes and corresponding class confidences directly while 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 to detect objects in various sizes. Since the prediction modules composed of $3 \times 3$ convolution filters detects bounding box on each layer separately, they can’t reflect appropriate contextual information. It causes a problem named as “Box-in-Box” as shown in Figure 1. In the figure, we can see that SSD often detects a single object with two overlapped boxes. The smaller box has partial image such as an upper body of a person or a head of an animal.

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 [7] and the deconvolutional layer are added before the prediction module. 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 improve generalization performance.
2. Related Works

Overfeat [16], SPPNet [6], R-CNN [5], Fast R-CNN [4], Faster R-CNN [14] and R-FCN [10] 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 problems.

Redmon et al. proposed a method named as YOLO [12] to predict bounding boxes and associated class probabilities in a single step by framing object detection as a regression problem. However, since YOLO [12] uses only the highest-level feature maps to detect objects, there is a lack of information on the small objects, which results in a somewhat inaccurate detection especially for small objects.

In order to solve this problem, SSD [11] utilized not only the highest-level layer but also lower layers which have enough resolution to detect small objects. As mentioned in Inside-Outside Net (ION) [17] and HyperNet [2], 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 [11], 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. However, as mentioned in MS-CNN [9], 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 use 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 [11]. First, DSSD [3] could obtain higher accuracy by changing the base network to Residual-101 [7] and combining the deconvolutional layers with the existing multiple layers to reflect the large scale context. However, with the use of deep structure of Residual-101 [7] and deconvolutional layers, the processing speed is very slow, which prohibits the method to be used for real-time detection problems. In addition, Ren et al. [13] introduced a recurrent

rolling convolution (RRC) architecture to improve detection performance by mutually complementing layers having different sizes of contextual information. RRC [13] 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 [13], Rainbow SSD (R-SSD) [8] 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.

3. Residual Feature and Unified Prediction Network

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. The proposed architecture is named as RUN which is an abbreviation for Residual feature maps for Unified prediction Network.

3.1. Residual Feature Maps

Recent CNN models designed for object detection make 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:

$$x_n = F_n(x_{n-1}) = (F_n \circ F_{n-1} \circ \cdots \circ F_1)(I)$$

$$\text{Scores} = P(x_n),$$

where I is an input image, $x_n$ is the $n^{th}$-level feature map, $P$ is a prediction function, and $F_n$ is a combination of non-linear 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 as $k$ becomes smaller.

SSD [11] applies several feature maps with different scales directly as an input to the prediction module to calculate object positions and classification scores, which can be denoted by the following equation:

$$\text{Detection} = \{P_1(x_{s_1}), P_2(x_{s_2}), \ldots, P_k(x_{s_k})\},$$

where $s_1$ to $s_k$ are feature indices for source feature map for multi-scale prediction, $P_k$ is a function that outputs multiple

---

1In this paper, the term level is used interchangeably with layer. Highest level indicates the the farthest layer from the input layer.
Figure 2: Networks of SSD and RUN. Top: SSD. Bottom: RUN. Compared with SSD, RUN has residual blocks and unified prediction module. The arrow from the bottom to the top indicates the deconvolutional branch.

objects with different positions and scores. Combining (1) and (3), it can be expressed as

\[
\text{Detection} = \{ P_1(x_{s1}), P_2(F_{s1}^2(x_{s1})), \ldots, P_k(F_{s1}^n(x_{s1})) \}, \quad (4)
\]

where \( F_n^0(x_n) = (F_b \circ \ldots \circ F_{n+1})(x_n) \). Here, the earlier feature map \( x_{s1} \) needs to learn high-level abstraction to improve the performance of \( P_1(x_{s1}) \). 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.

To resolve this problem, SSD [11] added L2 normalization layer between the conv4_3 layer and the prediction module to reduce the magnitude of the gradients from the prediction module. Cai et al. [2] tried to solve this problem by adding a convolutional 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 as shown in Figure 2. 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 architecture of the proposed multi-way ResBlocks are shown in Figure 3. Convolutional 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. It helps to improve the connected feature structure of the SSD [11] by forcing it not to learn high-level abstraction.

Also, the depths of the earlier layers (eg. conv4_3) used for small-sized object detection in SSD are very shallow. Therefore, in SSD, small objects can not 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 \(3 \times 3\) convolution layer in branch2 of the ResBlock as shown in Figure 3 to reflect the peripheral contextual information.

Branch3 in the right side of Figure 3 contains a deconvolutional layer whose input is the feature maps of the consecutive layer. This is similar to a structure proposed in [3] and [13], 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 case of detecting only a part of the actual object, not the entire object. Thus, it can be a remedy for the box-in-box problem described earlier. The effect of this is intuitively shown in the right side of Figure 1. Finally, the proposed architecture in Figure 2 can be expressed as follows:

\[
\text{Detection} = \{ P_1(\hat{x}_{s_1, s_2}), P_2(\hat{x}_{s_2, s_3}), \ldots, P_{k-1}(\hat{x}_{s_{k-1}, s_k}), P_k(\hat{x}_{s_k}) \}, \quad (5)
\]

where \(\hat{x}_{a,b} = B_1(x_a) + B_2(x_a) + B_3(x_b)\) and \(\hat{x}_a = B_1(x_a) + B_2(x_a)\). Here, \(B_1, B_2\) and \(B_3\) indicate branch1, branch2 and branch3, respectively.

### 3.2. Unified Prediction Module

In the previous methods using multi-scale features, feature maps in each scale differed not only in length but also in the underlying contextual information. Therefore, it was natural to learn the prediction modules separately for each scale. While we propose residual feature maps, we enforce all the feature maps to have the same depth of 256 as shown in Fig. 3. Thus, structurally, it is possible to unify convolutional layers of different prediction modules by sharing their weights. The idea of the unified prediction module is similar to [8], but our method is different from R-SSD [8] in information of input feature maps.

This approach has a couple of positive effects. First, it makes differently-scaled feature maps have similar level of information. Second, it has a regularization effect to avoid over-fitting in the training. In multi-scale detection, the earliest feature maps are obtained from much shallower layers than the top feature maps. Thus, it causes over-fitting and mis-detection for small objects. Unified prediction in combination with the residual feature block makes the earliest feature maps rich in context.

### 4. Experiment

We experimented the proposed method on PASCAL VOC 2007 and PASCAL VOC 2012 dataset. First, we tested whether each of the proposed ResBlock components (branches) is meaningful with PASCAL VOC 2007 test set. And then, we compared the performance of the combination of components with existing methods using PASCAL VOC 2012 test set. Our implementation is based on the publicly available SSD\(^2\) Reduced VGG-16 model pretrained on the ILSVRC CLS-LOC dataset [15] is used as the backbone of both SSD and ours.

| Method | mAP |
|--------|-----|
| SSD 300 | 77.5 |
| SSD 300 + 2WAY | 78.3 |
| SSD 300 + 2WAY + Unified Pred | 78.4 |
| SSD 300 + 3WAY | 78.8 |
| SSD 300 + 3WAY + Unified Pred | 79.1 |

Table 1: PASCAL 2007 test detection results.

#### Training

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 120k iterations, respectively. The training was terminated at 140k iterations. For the 3-way model, we froze all the weights

\(^2\)https://github.com/weiliu89/caffe/tree/ssd
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 method.

### PASCAL VOC 2007

We trained our model on VOC 2007 trainval set. Table 1 shows our result on PASCAL VOC 2007 test set. In the table, SSD300 is the latest SSD results with data augmentation mentioned in [3]. 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, the 3-way models were fine-tuned on each 2-way model. In this experiment, we observed that the proposed model with only 2-way residual block without deconvolution path achieved 0.8% higher mAP than that of SSD. The 3-way model which further utilizes deconvolutional layers was up to 0.7% higher than 2-way model. The unified prediction module made better advance in 3-way model than 2-way model, which scored 79.1% and 78.4% respectively.

### PASCAL VOC 2012

For VOC 2012 test, we trained model on 07++12 dataset consisting 07 trainval, 07 test and 12 trainval. First, we performed an experiment applying 2-way ResBlock in combination with the unified prediction, then, another experiment was performed using 3-way ResBlock with unified prediction after freezing the weights of the contained 2-way ResBlock.

Table 2 shows our results on VOC 2012 test with SSD 300 and other models based on SSD 300. The time column shows network forward time using Titan X Pascal GPU, cuDNN v5.1 and Intel I7-6700k@4.0GHz. The time for DSSD is blank since its implementation is not available. Compared with SSD300, our 2-way and 3-way models require 25% and 50% more computation time, respectively. However, ours has higher mAP than the others. Especially, 3-way model which achieves 77.9% mAP outperforms other SSD-based models.

### 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 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 multi-scale 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. We expect the proposed method be not restricted to SSD-based methods but also applicable to other structures utilizing multi-scale features.

### References

[1] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick. Inside-outside 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 multi-scale deep convolutional neural network for fast object detection. In *ECCV*, 2016.

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

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

[5] 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.

[6] 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.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Con-

| Method     | data   | Time(s) | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  | mAP  |
|------------|--------|---------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| SSD300     | 07++12 | 10.4    |       |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| DSSD       | 07++12 | N/A     |       |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| RUN300 2W | 07++12 | 18.2    | 76.4  | 76.3 | 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 | 35.3 | 79.6 | 75.7 | 85.2 | 73.9 | 88.1 | 82.9 | 74.4 | 61.9 | 47.6 | 82.7 |
| RUN300 3W | 07++12 | 15.6    | 77.0  | 89.3 | 84.2 | 75.1 | 63.6 | 51.0 | 83.8 | 91.6 | 59.5 | 82.0 | 64.2 | 90.0 | 86.4 | 86.2 | 82.9 | 52.2 | 82.0 | 73.4 | 87.7 | 74.6 | 58.2 | 80.7 | 64.6 | 90.0 | 91.6 | 59.5 | 82.0 |

Table 2: PASCAL 2012 test detection results. 07++12: 07 trainval + 07 test + 12 trainval. Time: net. forward time (millisecond)
[8] J. Jeong, H. Park, and N. Kwak. Enhancement of ssd by concatenating feature maps for object detection. *arXiv preprint arXiv:1705.09587*, 2017.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] O. 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.

[16] 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.
Figure 5: Detection examples of RUN300 3-way on PASCAL VOC 2012 test set compared with SSD300 model. For each pair, the left side is the result of SSD and right side is the result of RUN. We show detections with scores higher than 0.6. Each color corresponds to an object category.