MFPN: A NOVEL MIXTURE FEATURE PYRAMID NETWORK OF MULTIPLE ARCHITECTURES FOR OBJECT DETECTION

Tingting Liang[1], *Yongtao Wang*[1], *Qijie Zhao*[1], *huan zhang*[1], *Zhi Tang*[1], *Haibin Ling*[2]

[1]Wangxuan Institute of Computer Technology, Peking University
[2]Department of Computer Science , Stony Brook University
{tingtingliang, wyt, zhaoqijie, zhanghuan666, tangzhi}@pku.edu.cn
hling@cs.stonybrook.edu

ABSTRACT

Feature pyramids are widely exploited in many detectors to solve the scale variation problem for object detection. In this paper, we first investigate the Feature Pyramid Network (FPN) architectures and briefly categorize them into three typical fashions: top-down, bottom-up and fusing-splitting, which have their own merits for detecting small objects, large objects, and medium-sized objects, respectively. Further, we design three FPNs of different architectures and propose a novel Mixture Feature Pyramid Network (MFPN) which inherits the merits of all these three kinds of FPNs, by assembling the three kinds of FPNs in a parallel multi-branch architecture and mixing the features. MFPN can significantly enhance both one-stage and two-stage FPN-based detectors with about 2 percent Average Precision(AP) increment on the MS-COCO benchmark, at little sacrifice in running time latency. By simply assembling MFPN with the one-stage and two-stage baseline detectors, we achieve competitive single-model detection results on the COCO detection benchmark without bells and whistles.

Index Terms— Object Detection, Feature Pyramid Network, Scale Variation

1. INTRODUCTION

Object detection is a fundamental research topic in image/video understanding. It can serve as a prerequisite for various image/video retrieval, intelligent surveillance and autonomous driving. Existing deep learning-based detectors can be briefly categorized into two branches: one-stage detectors such as SSD [1], RefineDet [2] and RetinaNet [3], which utilize CNN directly to predict the bounding boxes; and two-stage methods including Faster R-CNN [4], R-FCN [5] and Mask R-CNN [6], which generate a set of candidate proposals and then exploit the extracted region features from CNN for further refinement. Although encouraging progresses have been made, the existing detectors are still suffering from the problems caused by the scale variation across object instances. An intuitive approach to solve the scale variation problem is to use a multi-scale image pyramid [7]. However, the dramatic increase in inference time makes the image pyramid methods infeasible for practical applications. Other kinds of methods [8][9][10][1] aim to employ the feature pyramid within the network, to approximate the image pyramid at a lower computational cost. Feature Pyramid Network (FPN) [8] is the most representative one, which incorporates high-semantic information in both high-level and low-level features with a top-down pathway, achieving superior performance. However, this top-down architecture design has the following intrinsic limitations: (1) it only introduces high-semantics information from deep layer to shallow layer, but does not consider the assistance of shallow layer to deep layer; (2) the top-down architecture makes the features of small objects largely depend on the features of larger objects, and this dependence is not always beneficial. For instance, we conduct a toy experiment by change the number of FPN channels in the baseline detector RetinaNet-ResNet50 (input size 800) [3] to test the accuracy bottleneck, and the results are shown in Figure 1. It is notable that when the channel dimension increases to 768, the accuracy growth is negligible with a lot of additional computation and parameters. This experiment demonstrates that such a top-down FPN architecture has bottleneck restrictions.

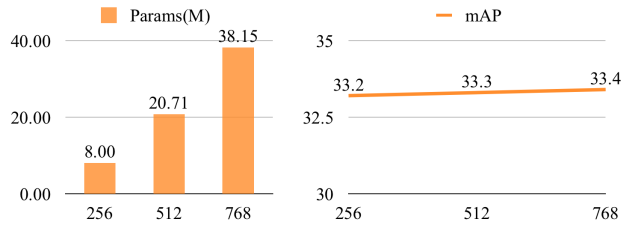

Fig. 1. Left: the numbers of parameters with different numbers of channels of FPN. Right: the detection accuracies with different numbers of channels of FPN. The baseline detector is RetinaNet500-ResNet50.

To address these problems, we rethink the feature pyramid network and summarize the architectures of FPN with three

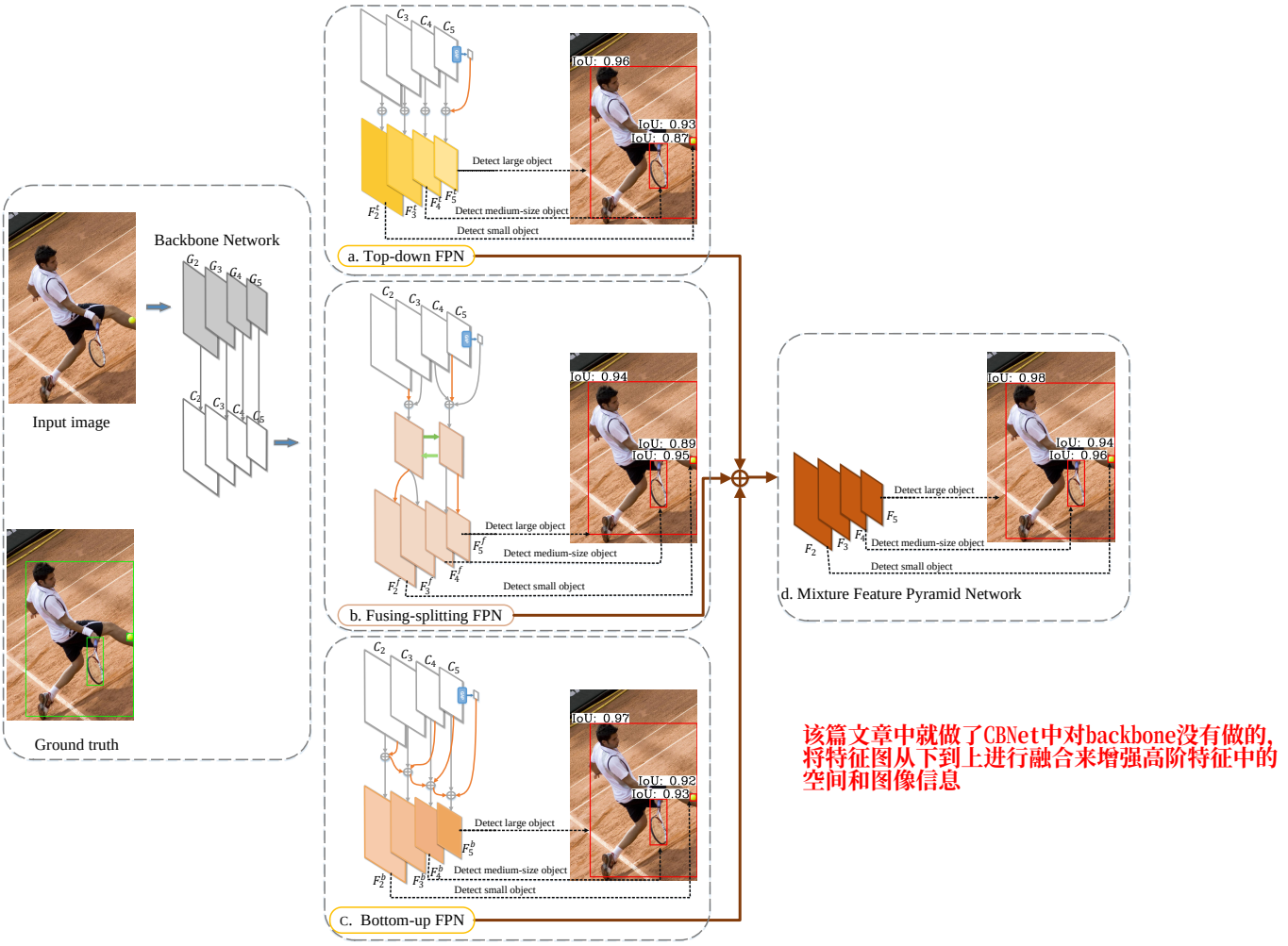

Fig. 2. Exsample results of object detectors using feature pyramid networks of different architectures (the baseline detector is RetinaNet500-ResNet50). Our MFPN performs best: detecting objects of small-size, medium-size and large-size with the highest IoU. Green boxes: *ground truth*, Red boxes: *detection result*.

different fashions: top-down, fusing-splitting and bottom-up. As illustrated in Figure 2 from top to bottom, we design an instance FPN for each FPN architecture. The Top-down FPN is an improved version of the original FPN [8], which introduces high-level semantic contexts to low-level features for better detecting small objects. In particular, we newly propose the bottom-up FPN, which introduces low-level details to high-level features, helping the high-level features obtain more spatial information thus can better detect large objects. Deviated from the interdependent relationship between deep and shallow features, we propose a novel Fusing-splitting FPN, which first fuses higher-level and lower-level features and then splits the fused feature into multi-scale features. Further, as illustrated in Figure 2, we propose a novel feature pyramid network that assembles these three FPNs of different architectures, named Mixture Feature Pyramid Network (MFPN). Experimental results show that the proposed MFPN can significantly enhance these FPN based detectors by about 2 percent Average Precision(AP), and can improve the detection performance of objects of all scale ranges (e.g., as depicted in Figure 2). Moreover, competitive single-model detection results are achieved by both one-stage and two-stage baseline detectors equipped with MFPN.

In summary, our main contributions are as follows:

- We design three FPNs of different architectures, Top-down FPN, Bottom-up FPN, and Fusing-splitting FPN, which have better detection performance for small objects, large objects, medium-size objects respectively.
- We propose a novel Mixture Feature Pyramid Network (MFPN) which inherits all the merits of the three FPNs, by assembling them in a parallel multi-branch architecture and mixing the features extracted by each branch.
- We achieve significant better detection results than both

one-stage and two-stage FPN-based detectors on MS COCO benchmark.

2. RELATED WORK

Addressing scale variation issue is critical for object detection, segmentation and other tasks that require predictive location[7]. To tackle the scale variation problem, an intuitive way is to use a multi-scale image pyramid during training and inference [5][11][12]. Different from methods with fixed or random scale transform, SNIP [7] selectively back-propagates the gradients of object instances of different sizes as a function of image scale. In addition, SNIPER [13] samples low-resolution chips to accelerate multi-scale training. Multi-scale image pyramid greatly improves accuracy, but suffers a lot from increasing inference time.

The feature pyramid method, that is, constructing and using the feature pyramid within the network, is more widely used to deal with scale variation, due to its lower computation cost. Methods like SSD [1] and MS-CNN [14] directly perform small objects detection on higher resolution feature maps while large ones on lower resolution feature maps extracted by the backbone network (e.g., VGG). Due to the backbone networks are originally designed for classification task, directly using the features extracted by them leads to suboptimal performance. Hence, some recent works try to alleviate this problem by enhancing the features extracted by backbones with novel feature enhancement modules, e.g., RFBNet [15] and TridentNet [16]. Feature Pyramid Networks (FPN) [8] is commonly exploited by state-of-the-art object detectors, e.g., Mask RCNN [6], RetinaNet[3], RefineDet [2], etc., which proposes a subnet with top-down architecture to construct feature pyramid. Recently, Multi-level FPN[17] introduces multiple U-shape modules after a backbone network to extract multi-level pyramidal features, and builds a powerful one-stage detector. Libra R-CNN [18] and [19] are two recently proposed feature pyramid networks of Fusing-splitting architecture, who combine features of all scales and then generate features at each scale by a global attention operation on the combined features. As stated in section 1, these FPNs have their own intrinsic limitations since they are designed with only one specific kind of FPN architecture (i.e., top-down, or fusing-splitting, or bottom-up).

3. PROPOSED METHOD

In this work, we first introduce three kinds of FPN architectures, that is, Top-down, Bottom-up and Fusing-splitting. As illustrated in Figure 2, each pyramidal feature map (denoted as G_2, G_3, G_4, G_5) extracted by the backbone is followed by an extra 1×1 convolution. Then, these feature maps (denoted C_2, C_3, C_4, C_5) are used to build feature pyramid for object detection by each FPN of different architectures as following.

3.1. Top-down FPN

The major characteristic of top-down FPN architecture is: the FPN feature maps (denoted as $F_2^t, F_3^t, F_4^t, F_5^t$) are sequentially constructed in a top-down manner, that is, the smaller scale (higher-level) feature map is constructed first. we adopt the most widely used top-down architecture FPN[8] with some modifications. To be more specific, we plug an extra global average pooling(GAP)[20] layer above the deepest layer of the backbone to extract the global context, i.e., G5. Moreover, GAP can learn richer semantic information and highlights the discriminative object regions detected by CNNs[21], thus can propagate more semantic information to the larger scale(lower-level) feature maps. Same as the original FPN[8], each feature map (F_i^t) of Top-down FPN is iteratively built by combining the same level backbone feature map (C_i) and the higher-level FPN feature map (F_{i+1}^t):

$$F_i^t = \mathbf{W^t}_{i+1} \otimes (U(F_{i+1}^t) + C_i), \tag{1}$$

where $U(\cdot)$ denotes the upsample operation with a factor of 2 and $\mathbf{W_i^t}$ is a 3×3 convolution filter. Since the top-down architecture iteratively propagates semantic information of higher-level backbone features to the more detailed lower-level FPN feature maps, it is better at detecting small objects.

3.2. Bottom-up FPN

Contrary to the top-down architecture, the major characteristic of bottom-up FPN is: the FPN feature maps (denoted as $F_2^b, F_3^b, F_4^b, F_5^b$) are sequentially constructed in a bottom-up manner, that is, the large scale (lower-level) feature map is constructed first. As illustrated in Figure 2.c, each feature map (F_i^b) of the Bottom-up FPN is obtained by merging the same level backbone feature map (C_i), the backbone feature map (C_{i+1}) above it, and the FPN feature map (F_{i-1}^b) below it, which can be formulated as:

$$F_i^b = \mathbf{W_i^b} \otimes (D(F_{i-1}^b) + C_i + U(C_{i+1})), \tag{2}$$

where $D(\cdot)$ denotes MaxPool operation with a factor of 2 and $\mathbf{W_i^b}$ is a 3×3 convolution filter. Because the bottom-up architecture propagates the spatial detail information of lower-level backbone features to the higher-level FPN features, it is better at detecting large objects. Obviously, Bottom-up FPN and Top-down FPN are complementary to each other.

3.3. Fusing-splitting FPN

Since the feature maps of the Top-down FPN and Bottom-up FPN are sequentially built, the earlier constructed features always affect the subsequent ones, and this interdependent design may lead to some intrinsic limitation. To address this problem, we design a Fusing-splitting FPN, which first combines the higher-level and lower-level backbone features, and

then splitting the combined features to multi-scale FPN features. In practice, the highest two backbone feature maps are

Table 1. Object detection result comparison on COCO minival for the three FPNs of different architectures and the proposed MFPN. The baseline is RetinaNet500-ResNet50.

Method	**Parameters(M)**	**AP**	AP_s	AP_m	AP_l
FPN (Baseline)	8.00	33.2	15.0	37.5	47.4
Top-down	8.52	33.5	**15.2**	38.1	47.6
Bottom-up	8.52	33.5	14.4	37.9	**48.7**
Fusing-splitting	6.49	33.6	14.7	**38.5**	48.1
MFPN	11.47	**34.8**	**16.8**	**39.1**	**49.0**

Table 2. Object detection results comparison on COCO minival for different combinations of the three kinds of FPN architectures. The baseline is RetinaNet500-ResNet50.

Method	**AP**	AP_s	AP_m	AP_l
Baseline	33.2	15.0	37.5	47.4
Bottom-up + Fusing-splitting	34.3	16.0	39.0	48.6
Top-down + Bottom-up	34.3	15.8	38.9	48.9
Top-down + Fusing-splitting	33.8	15.7	38.4	47.7
MFPN	**34.8**	**16.8**	**39.1**	**49.0**

merged into a combined feature map α_s, and the lowest two backbone feature maps are merged into α_l:

$$\alpha_s = C_4 + U(C_5), \alpha_l = D(C_2) + C_3. \tag{3}$$

After obtaining the first-round combined features, we further fuse them as following,

$$\begin{aligned} \beta_s &= \mathbf{W_s^f} \otimes cat(\alpha_s, D(\alpha_l)), \\ \beta_l &= \mathbf{W_l^f} \otimes cat(U(\alpha_s), \alpha_l), \end{aligned} \tag{4}$$

where $\mathbf{W_s^f}$ and $\mathbf{W_l^f}$ are two 3×3 convolution filters, and $cat(\cdot)$ represents concat operation along channel dimension. After these operations, feature maps β_s, β_l have fused informations from all level features. Finally, we simply resize β_s, β_l into multi-scale pyramidal feature maps, that is,

$$\begin{aligned} F_2^f &= U(\beta_l), F_3^f = \beta_l; \\ F_4^f &= \beta_s, F_5^f = D(\beta_s). \end{aligned} \tag{5}$$

By the above two rounds fusing and the splitting operations, all the feature maps of the Fusing-splitting FPN incorporate information from the backbone feature maps of all levels. Moreover, *the two medium-scale feature maps* (F_3^f and F_4^f) are obtained with less downsampling or upsampling operation. Hence, Fusing-splitting FPN has a stable improvement in detecting medium-sized objects.

Table 3. Performance comparison between FPN and MFPN on the COCO minival. R: ResNet. X: ResNext-101-64x4d.

Baseline	**Method**	AP	AP_S	AP_M	AP_L	**time(ms)**
Retinanet-R50	FPN	35.6	20.0	39.6	46.8	85
	ours	37.9	21.4	41.9	49.7	86
Retinanet-X101	FPN	40.0	23.0	44.3	52.7	196
	ours	42.1	24.9	46.8	55.3	196
Faster R-CNN -R50	FPN	36.4	21.5	40.0	46.6	82
	ours	38.6	22.6	42.8	49.7	93
Cascade Mask R-CNN-R101	FPN	42.7	23.8	46.5	56.9	196
	ours	44.4	25.9	48.1	58.2	204

3.4. Mixture Feature Pyramid Network (MFPN)

Now we propose a more powerful feature pyramid network named MFPN by integrating the above three FPNs. Intuitively, MFPN inherits all the merits of the three FPNs and performs better to handle scale variation problem in object detection. By integrating the three FPNs in one network, we can avoid a large increase in the number of parameters by sharing one backbone network. The network architecture of MFPN is illustrated in Figure 2, each feature map of MFPN is obtained by summing the same level feature map of the three feature pyramids along spatial dimension, that is,

$$F_i = F_i^t + F_i^b + F_i^f, i = 2, 3, 4, 5. \tag{6}$$

MFPN can play all the roles played by FPN, including as anchor feature to improve the accuracy[3], or as neck feature to boost RPN[4] for better candidate proposals and connect with RoI Extractor[4][5][6]for better RoI features.

4. EXPERIMENT

4.1. Dataset and Implementation details

Dataset Description. We present experimental results on the bounding boxes detection task of the challenging *MS-COCO* benchmark [25]. For training, validation and testing processes, we follow [2] and [8], train on the union of 11.8k training images(including the 80k `train` split and a random 35k subset of images from the 40k image `val` split), conduct ablation study on 5k `minival` split for convenience. Then, to compare the accuracy with state-of-the-art FPN-based methods, we report results of `test-dev` split images.

Implementation Details. The backbones used in this paper are all pre-trained on ImageNet [26]. For ablation study experiments, we train detectors 12 epochs in total, with learning rate starting from 0.02 and the batch size is 16. Cascade Mask R-CNN-MFPN and RetinaNet-X101-MFPN are trained for 20 epochs and the initial learning rate is set to 0.01. For evaluation, detectors run on a single Titan X GPU with CUDA 9 and CUDNN 7, with a batch size of 1.

Table 4. Detection accuracy comparisons with the state-of-the-art FPN-based methods on *MS-COCO* `test-dev` set.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
one-stage:							
SSD512 [1]	VGG-16	28.8	48.5	30.3	10.9	31.8	43.5
RefineDet512 [2]	ResNet-101	36.4	57.5	39.5	16.6	39.9	51.4
RetinaNet800 [3]	Res101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
CornerNet [22]	Hourglass-104	40.5	56.5	43.1	19.4	42.7	53.9
M2Det [17]	VGG-16	41.0	59.7	45.0	22.1	46.5	53.8
FSAF [23]	ResNext-101-64x4d	42.9	63.8	46.3	26.6	46.2	52.7
two-stage:							
Faster R-CNN w FPN [8]	ResNet101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Deformable R-FCN [11]	Inc-Res-v2	37.5	58.0	40.8	19.4	40.1	52.5
Mask R-CNN [6]	ResNeXt-101	39.8	62.3	43.4	22.1	43.2	51.2
TridentNet [16]	ResNet-101-Deformable	42.7	63.6	46.5	23.9	46.6	56.6
Cascade R-CNN [24]	ResNet101-FPN	42.8	62.1	46.3	23.7	45.5	55.2
SNIP [7]	ResNet-101-Deformable	44.4	66.2	44.9	27.3	47.4	56.9
SNIPER [13]	ResNet-101-Deformable	46.1	67.0	51.6	29.6	48.9	58.1
Ours:							
MFPN-Cascade Mask R-CNN	ResNext-101-64x4d	47.6	66.7	52.0	29.4	50.8	59.6
MFPN-RetinaNet	ResNext-101-64x4d	43.4	63.4	46.5	26.1	47.3	54.0

4.2. Ablation Studies

Compare the three FPNs As shown in Table 1, Top-down

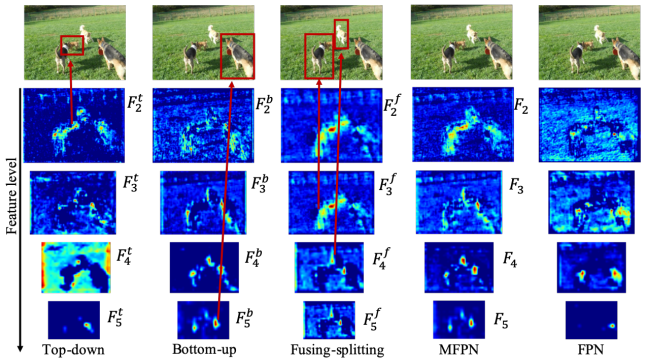

Fig. 3. Heatmap visualization exsamples of MFPN and FPN.

FPN gets the highest score for small objects(AP_s of 15.2), while Bottom-up FPN wins for large objects(AP_l of 48.7) and Fusing-splitting FPN is best at detecting medium-sized objects(AP_m of 38.5). When we add up the three FPNs, the overall AP is 1.5 higher than FPN. We also conduct experiments of multiple combinations of Top-down FPN, Bottom-up FPN and Fusing-splitting FPN in Table 2. The combination of Top-down and Bottom-up gets the highest result (36.8) among the pair-wise combinations. At the same time, to further improve the accuracy of AP_{75} and enhance the detection accuracy of hard samples, we adopt a combination of three FPNs. These results fully confirm our expectations and prove that our design is reasonable and effective.

MFPN can significantly enhance FPN-based detectors We further evaluate the proposed MFPN with different backbones and detectors, using input image scale of 800 pixels. Results are detailed in Table 3. MFPN consistently improves the detection accuracy for various backbones. For MFPN-Retinanet and MFPN-Faster R-CNN, we adopt balanced loss[18] instead of smooth L1 to better handle sample imbalance problem. Our MFPN introduces marginal computation cost to the whole detection network, leading to negligible loss of inference speed. Especially, we improve RetinaNet by 2.1 AP on Retinanet ResNeXt-101 without additional inference latency increment, and 1.6 percent of AP on Cascade Mask RCNN-ResNet 101 with only 8ms latency increment.

MFPN can learn better features for object detection To verify that the proposed MFPN can learn effective feature for detecting objects of various sizes, we visualize the activation values of the output of FPN and MFPN along scale and level dimensions, such an example shown in Figure 3. The input image contains four dogs with different sizes. We can find that: 1) For detecting the smallest dog, the lowest feature from Top-down FPN F_2^t achieves clearer and noise-free semantics than that from FPN. 2) Compared with FPN, Bottom-up FPN obtains better high-level FPN features with three clear activation points in F_5^b, and can better detecting the biggest dog. 3) F_3^f , F_4^f from Fusing-splitting FPN have larger activation regions than FPN, containing more detailed information, thus cann better detecting the two medium-sized dogs. 4) The responses of MFPN to objects are accurate, while the ones of FPN are hindered by meaningless noise. This implies: 1) MFPN is good at learning the characteristics of objects. 2) It is necessary to use MFPN to detect objects of various sizes.

4.3. Compare with state-of-the-art FPN-based methods

We evaluate MFPN on the *COCO* `test-dev` set and compare it with recent state-of-the-art FPN-based methods. The model is trained using scale jitter over scales $\{640, 672, 704, 736, 768, 800\}$. For fair comparison, we only compare the results produced from single models without ensemble or multi-scale testing. As shown in Table 4, MFPN based detectors, RetinaNet-MFPN, and Cascade Mask R-CNN-MFPN, achieve superior results without bells and whistles. RetinaNet-MFPN gets AP (**43.4**), which surpasses all other one-stage detectors. Cascade Mask R-CNN-MFPN obtains AP of **47.6**, outperforms TridentNet, SNIP and SNIPER, who uses image pyramid training and testing strategies. In conclude, MFPN is compatible with both powerful one-stage detectors and two-stage detectors and can achieve very competitive single-model results.

5. CONCLUSION

In this paper, we first describe three FPNs of different architectures(*i.e.,*, Top-down, Bottom-up, and Fusing-splitting) for extracting multi-scale features to solve the scale variation problem for object detection. Based on them, we propose a novel Mixture Feature Pyramid Network(MFPN), which is effective for learning powerful multi-scale features and can be simply assembled into both one-stage detectors and two-stage detectors. On the MS-COCO benchmark, MFPN improves the performance for all scale-ranges and enhances both one-stage and two-stage FPN-based detectors with 2 % AP increment, which leads to very competitive results.

6. REFERENCES

[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and Alexander C. Berg, "SSD: Single shot multibox detector," in *ECCV*, 2016.

[2] S. Zhang, L.n Wen, X. Bian, Z. Lei, and S. Li, "Single-shot refinement neural network for object detection," in *CVPR*, 2018.

[3] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *ICCV*, 2017.

[4] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[5] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *NIPS*, 2016.

[6] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *ICCV*, 2017.

[7] S. Bharat and D. Larry S, "An analysis of scale invariance in object detection–snip," in *CVPR*, 2018.

[8] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017.

[9] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta, "Beyond skip connections: Top-down modulation for object detection," in *CoRR*, 2016.

[10] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: reverse connection with objectness prior networks for object detection," in *CVPR*, 2017.

[11] J. Dai, H. Qi, Y. Xiong, Yi. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *ICCV*, 2017.

[12] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *CVPR*, 2017.

[13] B. Singh, M. Najibi, and L. S. Davis, "Sniper: Efficient multi-scale training," in *NIPS*, 2018.

[14] Z. Cai, Q. Fan, R. Schmidt Feris, and N. Vasconcelos, "A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection," in *ECCV*, 2016.

[15] S. Liu, D. Huang, and Y. Wang, "Receptive field block net for accurate and fast object detection," in *ECCV*, 2018.

[16] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-Aware Trident Networks for Object Detection," in *ICCV*, 2019.

[17] Q. Zhao, T. Sheng, Y. Wang, and Z. Tang, "M2det: A single-shot object detector based on multi-level feature pyramid network," in *AAAI*, 2019.

[18] J. Pang, K. Chen, J. Shi, Hu. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards Balanced Learning for Object Detection," in *CVPR*, 2019.

[19] T.o Kong, F. Sun, W. Huang, and H. Liu, "Deep Feature Pyramid Reconfiguration for Object Detection," in *ECCV*, 2018.

[20] M. Lin, Q. Chen, and S. Yan, "Network in network," in *ICLR*, 2014.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[22] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *ECCV*, 2018.

[23] Chenchen Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *CVPR*, 2019.

[24] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *CVPR*, 2018.

[25] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, ," .

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," in *IJCV*, 2015.