AugFPN: Improving Multi-scale Feature Learning for Object Detection

Chaoxu Guo[1,2] Bin Fan[1,4*] Qian Zhang[3] Shiming Xiang[1,2] Chunhong Pan[1]
[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences [3]Horizon Robotics
[4]School of Automation and Electrical Engineering, University of Science and Technology Beijing
{*chaoxu.guo,smxiang,chpan*}*@nlpr.ia.ac.cn*,{*bin.fan*}*@ieee.org*, {*qian01.zhang*}*@horizon.ai*

Abstract

Current state-of-the-art detectors typically exploit feature pyramid to detect objects at different scales. Among them, FPN is one of the representative works that build a feature pyramid by multi-scale features summation. However, the design defects behind prevent the multi-scale features from being fully exploited. In this paper, we begin by first analyzing the design defects of feature pyramid in FPN, and then introduce a new feature pyramid architecture named AugFPN to address these problems. Specifically, AugFPN consists of three components: Consistent Supervision, Residual Feature Augmentation, and Soft RoI Selection. AugFPN narrows the semantic gaps between features of different scales before feature fusion through Consistent Supervision. In feature fusion, ratio-invariant context information is extracted by Residual Feature Augmentation to reduce the information loss of feature map at the highest pyramid level. Finally, Soft RoI Selection is employed to learn a better RoI feature adaptively after feature fusion. By replacing FPN with AugFPN in Faster R-CNN, our models achieve 2.3 and 1.6 points higher Average Precision (AP) when using ResNet50 and MobileNet-v2 as backbone respectively. Furthermore, AugFPN improves RetinaNet by 1.6 points AP and FCOS by 0.9 points AP when using ResNet50 as backbone. Codes are available on https://github.com/Gus-Guo/AugFPN.

1. Introduction

With the significant advances in deep convolutional networks (ConvNets), remarkable progress has been achieved in object detection. A number of detectors [10, 34, 9, 26, 31, 13, 22, 23, 12] have been proposed to steadily push forward the state-of-the-art. Among these detectors, FPN [22] is a simple and effective two-stage framework for object detection. Specifically, FPN builds a feature pyramid upon the inherent feature hierarchy in ConvNet by propagating the semantically strong features from high levels into features at lower levels.

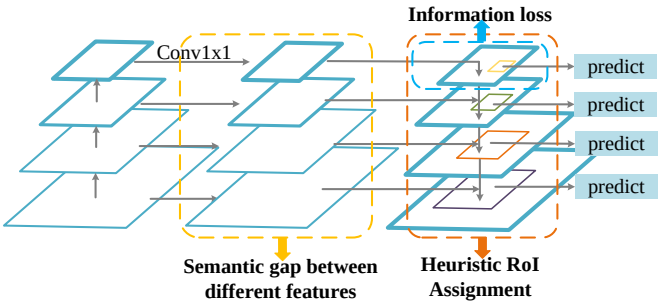


Figure 1. Three design defects in feature pyramid network: 1) **semantic gap between features at different levels** before feature summation, 2) **information loss** of the feature at the highest pyramid level, 3) **heuristic RoI assignment**.

By improving multi-scale features with strong semantics, the performance of object detection has been substantially improved. However, there exist some design defects within the feature pyramid in FPN, which is illustrated in Fig. 1. Basically, the feature pyramid in FPN can be formulated into three stages: (1) before feature fusion, (2) top-down feature fusion, and (3) after feature fusion. We find that each stage has an intrinsic flaw as described in the following:

Semantic gaps between features at different levels. Before performing feature fusion, features at different levels undergo a 1×1 convolution layer independently to reduce feature channels, where the large semantic gaps between these features are not considered. Fusing these features directly would degrade the power of multi-scale feature representation due to the inconsistent semantic information.

Information loss of the highest-level feature map. In feature fusion, features are propagated in a top-down path and low-level features can be improved with the stronger semantic information from higher-level features. Neverthe-

*Bin Fan is the corresponding author

less, the feature at the highest pyramid level instead loses information due to the reduced channels. The information loss can be mitigated by combining the global context feature [30] extracted by global pooling. But such a strategy of fusing the feature map into one single vector may lose the spatial relation and details because multiple objects may appear in one image.

Heuristical assignment strategy of RoIs . After feature fusion, each object proposal is refined based on the feature grids pooled from one feature level, which is chosen based on the scales of proposals heuristically. However, the ignored features from other levels may be beneficial for object classification or regression. Considering this problem, PANet [25] pools RoIs features from all pyramid levels and fuses them with the max operation after adapting them with independent fully connected layers. Nevertheless, the max fusion would ignore features with smaller responses that may be also helpful and still does not exploit the features at other levels fully. Meanwhile, the extra fully connected layers increase the model parameters significantly.

In this paper, we propose AugFPN, a simple yet effective feature pyramid that integrates three different components to deal with the problems above respectively. First, Consistent Supervision is proposed to make the feature maps after lateral connection contain similar semantic information by enforcing the same supervision signals on these feature maps. Second, ratio-invariant adaptive pooling is utilized to extract diverse context information, which could reduce information loss of the highest-level feature in feature pyramid in a residual way. We name this procedure as Residual Feature Augmentation. Third, Soft RoI Selection is introduced to better exploit RoI features from different pyramid levels and produce a better RoI feature for subsequent location refinement and classification.

Without bells and whistles, AugFPN based Faster R-CNN outperforms FPN based counterparts by 2.3 and 1.7 Average Precision (AP) when using ResNet50 and ResNet101 as backbone respectively. Furthermore, AugFPN improves the overall performance by 1.6 AP when the backbone is changed to MobileNet-V2, which is a lightweight and efficient network. AugFPN can also be extended to one-stage detectors with minor modifications. By replacing FPN with AugFPN, RetinaNet and FCOS are improved by 1.6 AP and 0.9 AP respectively, which manifests the generality of AugFPN.

We summarize our contributions as follows:

- We reveal three design defects in FPN that prevent the multi-scale features from being fully exploited.
- A new feature pyramid network named AugFPN is proposed to address these problems with Consistent Supervision, Residual Feature Augmentation, and Soft RoI Selection respectively.
- We evaluate AugFPN equipped with various detectors and backbones on MS COCO and it consistently brings significant improvements over FPN based detectors.

2. Related Work

Deep Object Detectors. Contemporary object detection methods almost follow two paradigms, two-stage and one-stage. As a seminal work of the two-stage detection methods [10, 9, 34, 4, 22, 1, 36, 20, 21, 29], R-CNN [10] first employs selective search [38] to generate region proposals and then refines these proposals by extracting region features through a convolutional network. To improve the training and inference speed, SPP [14] and Fast R-CNN [9] first extract feature map of the whole image and then generate region features with spatial pyramid pooling and RoI pooling respectively. Finally the region fetures are used to refine the proposals. Faster R-CNN [34] proposes a region proposal network and develops an end-to-end trainable detector, which promotes the performance significantly and speed-up the inference. To pursue scale-invariance in object detection, FPN [22] builds an in-network feature pyramid based on the inherent feature hierarchy of convolution network and makes predictions at different pyramid levels according to the scales of region proposals. RoI Align [13] brings great improvement in both object detection and instance segmentation by addressing the quantization problem of RoI pooling. Deformable network [5, 43] improve the performance of object detection significantly by modeling the geometry structure of objects. Cascade R-CNN [1] introduces a multi-stage refinement into Faster R-CNN and achieves more accurate predictions of object locations.

Contrary to two-stage detectors, one-stage detectors [26, 31, 6, 32, 23, 18, 24, 33, 40, 42] are more efficient yet less accurate. SSD [26] places anchor boxes densely on multi-scale features and make predictions based on these anchors. RetinaNet [23] utilizes a feature pyramid similar to FPN as backbone and introduces a novel focal loss to address the imbalance problem of easy and hard examples. ExtremeNet[42] models the problem of object detection as detecting four extreme points of the objects. These works make significant progress from different concerns. In this paper, we study a better exploitation of multi-scale features.

Deep Supervision. Deep supervision [16, 19, 41, 7] is a wildly used technique to tackle the common problem of gradient vanishing or enhance the feature representation of intermediate layers. Huang *et al.* [16] incorporate several classifiers with various resource demands into a single deep network by training it at different levels simultaneously. PSPNet [41] introduces an additional pixel-level loss on intermediate layers in order to reduce the optimizing difficulty. Recently Nas-FPN [7] attaches classifier and regression heads after all intermediate pyramid networks with a goal of achieving *anytime detection*. Contrary to these

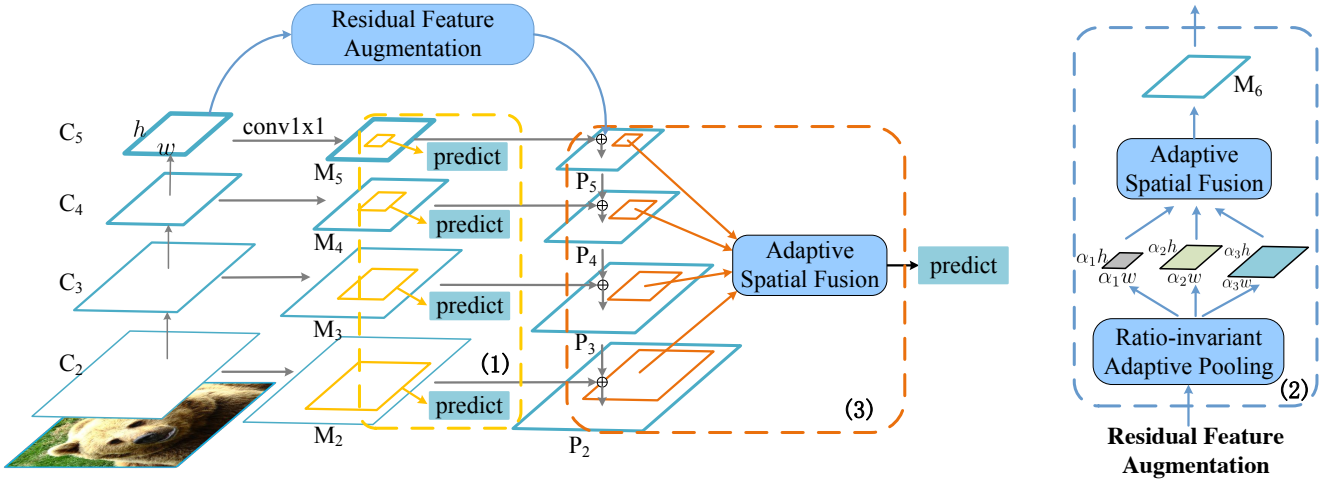


Figure 2. Overall pipeline of AugFPN based detector. (1)-(3) are three main components of AugFPN: Consistent Supervision, Residual Feature Augmentation, and Soft RoI Selection. For simplicity, the 3×3 convolution layers after feature summation are not shown.

works, we apply the instance-level supervision signals on features at all pyramid levels after lateral connection, aiming to narrow the semantic gaps between them and make the features more suitable for subsequent feature summation.

Context Exploitation. Several methods have proved the importance of context on both object detection [8, 30, 39] and segmentation [17, 27, 41]. Deeplab-v2 [3] proposes atrous convolution to extract multi-scale context and PSPNet [41] utilizes pyramid pooling to obtain hierarchical global context, both of which improve the quality of semantic segmentation greatly. Different from them, we perform ratio-invariant adaptive pooling to generate diverse spatial context information and utilize them to reduce information loss in channels of the feature at the highest pyramid level in a residual way.

Strategy of RoI Assignment. FPN [22] pools RoI features from one certain pyramid level, which is chosen according to the scales of RoIs. However, two proposals with a similar scale can be assigned to different feature levels under this strategy, which may produce sub-optimal results. To address this, PANet pools RoI features from all pyramid levels and fuses them by max operation after adapting them with fully connected layers. There is a distinct difference between PANet and our work that we propose a data-dependent way to generate adaptive weights and absorb features from all levels according to the weights. This enable the features at different levels to be better exploited. In addition, our work requires fewer parameters because no extra fully connected layers are required to adapt RoI features.

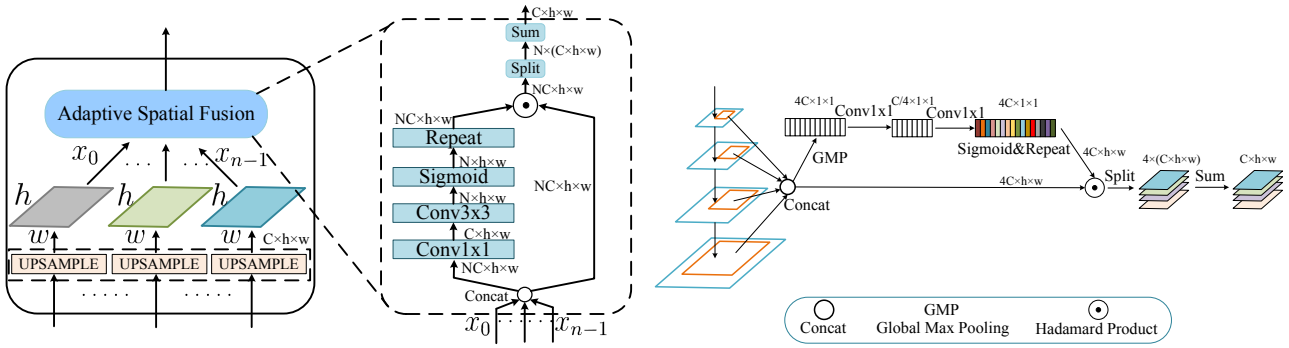
3. Methodology

The overall framework of AugFPN is shown in Figure 2. Following the setting of FPN [22], features used to build the feature pyramid are denoted as $\{C_2, C_3, C_4, C_5\}$, which correspond to the feature maps with strides $\{4, 8, 16, 32\}$ pixels in feature hierarchy w.r.t. the input image. $\{M_2, M_3, M_4, M_5\}$ are the features with reduced feature channels after lateral connection. $\{P_2, P_3, P_4, P_5\}$ are the features produced by feature pyramid. Three components of AugFPN will be discussed in the following subsections.

3.1. Consistent Supervision

FPN makes use of the in-network feature hierarchy that produces feature maps with different resolutions to build a feature pyramid. In order to integrate the multi-scale context information, FPN fuses features of different scales by upsampling and summation in a top down path. However, the features with different scales contain information at different abstract levels and there exist large semantic gaps between them. Although the method adopted by FPN is simple and effective, fusing multiple features with large semantic gaps would lead to a sub-optimal feature pyramid.

This inspires us to propose Consistent Supervision, which enforces the same supervision signals on the multi-scale features before fusion, with the goal of narrowing semantic gaps between them. Specifically, we first build a feature pyramid based on the multi-scale features $\{C_2, C_3, C_4, C_5\}$ from backbone. Then a Region Proposal Network (RPN) is appended to the resulting feature pyramid $\{P_2, P_3, P_4, P_5\}$ to generate numerous RoIs. To conduct Consistent Supervision, each RoI is mapped to all feature levels and the RoI features from each level of $\{M_2, M_3, M_4, M_5\}$ are obtained by RoI-Align [13]. After that, multiple classification and box regression heads are attached to these features to generate auxiliary loss. The parameters of these classification and regression heads are shared across different levels, which can further force different feature maps to learn similar semantic information



(a) (b)

Figure 3. (a) is the process of fusing different context features and structure of Adaptive Spatial Fusion. (b) is the details of Adaptive Channel Fusion. In (a), it depends on the sizes of inputs to decide whether to use upsample blocks in the dashed box.

besides the same supervision signals. For more stable optimization, a weight is used to balance the auxiliary loss generated by Consistent Supervision and the original loss. Formally, the final loss function of rcnn head is formulated as follows:

$$L_{rcnn} = \lambda \sum_{M=2}^{5} (L_{cls,M}(p_M, t^*) + \beta[t^* > 0] L_{loc,M}(d_M, b^*)) + \sum_{P=2}^{5} (L_{cls,P}(p, t^*) + \beta[t^* > 0] L_{loc,P}(d, b^*)). \quad (1)$$

$L_{cls,M}$ and $L_{loc,M}$ are objective functions corresponding to the auxiliary loss attached to $\{M_2, M_3, M_4, M_5\}$ while $L_{cls,P}$ and $L_{loc,P}$ are original loss functions on feature pyramid $\{P_2, P_3, P_4, P_5\}$. p_M, d_M and p, d are the prediction of intermediate layers and final pyramid layers respectively. t^* and b^* are the groundtruth class label and regression target respectively. λ is the weight used to balance the auxiliary loss and original loss. β is the weight used to balance classification and localization loss. The definition of $[t^* > 0]$ is as follows:

$$[t^* > 0] = \begin{cases} 1, t^* > 0 \\ 0, t^* = 0 \end{cases} \quad (2)$$

In the testing phase, the auxiliary branches are abandoned and only the branch after feature pyramid is utilized for final prediction. Therefore, Consistent Supervision introduces no extra parameters and computation to the model in inference.

3.2. Residual Feature Augmentation

In FPN, feature map at the highest level M_5 is propagated in a top-down path and fused with the feature maps at lower levels $\{M_4, M_3, M_2\}$ gradually. On the one hand, feature maps of lower levels are enhanced with the semantic information from higher levels and the resulting features are endowed with diverse context information naturally. On the other hand, M_5 suffers from the information loss due to the reduced feature channels and only contains single scale context information that is not compatible with the resulting features at other levels.

Based on this observation, we propose Residual Feature Augmentation to improve the feature representation of M_5 by utilizing a residual branch to instill diverse spatial context information into the original branch. We expect that the spatial context information can reduce the information loss in channels of M_5 and improves performance of the resulting feature pyramid simultaneously. To this end, we first produce multiple context features with different scales of $(\alpha_1 \times S, \alpha_2 \times S, .., \alpha_n \times S)$ by performing ratio-invariant adaptive pooling on C_5 whose scale is $S = h \times w$. Then each context feature undergoes a 1×1 convolution layer independently to reduce feature channel dimension to 256. Finally, they are upsampled to a scale of S via bilinear interpolation for subsequent fusion. Considering the aliasing effect caused by interpolation, we design a module named Adaptive Spatial Fusion (ASF) to adaptively combine these context features instead of simple summation. The detailed structure of ASF is illustrated in Figure 3(a). Specifically, ASF takes upsampled features as input and produces one spatial weight map for each feature. The weights are used to aggregate the context features into M_6, which is endowed with multi-scale context information.

After M_6 is generated by ASF, it is combined with M_5 by summation and propagated to fuse with other features at lower levels. Finally, a 3×3 convolution layer is appended to each feature map to construct a feature pyramid $\{P_2, P_3, P_4, P_5\}$.

Ratio-invariant adaptive pooling is different from PSP [41] in that PSP pools feature into multiple features with fixed sizes while ratio-invariant adaptive pooling takes the ratio of image into account, which is preferable to object detection. Furthermore, we fuse features with ASF instead of convolving the concatenated features directly , which is inferior as shown in the experiments in ablation study.

Method	Backbone	Schedule	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YoLOv2 [32]	DarkNet-19	-	21.6	44.0	19.2	5.0	22.4	35.5
SSD512 [26]	ResNet-101	-	31.2	50.4	33.3	10.2	34.5	49.8
RetinaNet [23]	ResNet-101-FPN	-	39.1	59.1	42.3	21.8	42.7	50.2
Faster R-CNN [22]	ResNet-101-FPN	-	36.2	59.1	42.3	21.8	42.7	50.2
Libra R-CNN [29]	ResNet-50-FPN	1x	38.7	59.9	42.0	22.5	41.1	48.7
Libra R-CNN [29]	ResNet-101-FPN	1x	40.3	61.3	43.9	22.9	43.1	51.0
Deformable R-FCN [4]	Inception-ResNet-v2	-	37.5	58.0	40.8	19.4	40.1	52.5
Mask R-CNN [13]	ResNet-101-FPN	-	38.2	60.3	41.7	20.1	41.1	50.2
Grid-R-CNN [28]	ResNet-101-FPN	2x	41.5	60.9	44.5	23.31	44.9	53.1
RetinaNet*	ResNet-50-FPN	1x	35.9	55.9	38.5	19.7	38.9	44.9
RetinaNet*	MobileNet-v2-FPN	1x	32.7	52.0	34.7	17.4	34.6	42.3
FCOS*	ResNet-50-FPN	1x	37.0	56.6	39.4	20.8	39.8	46.4
Faster R-CNN*	ResNet-50-FPN	1x	36.5	58.7	39.1	21.5	39.7	44.6
Faster R-CNN*	ResNet-101-FPN	1x	38.9	60.9	42.3	22.4	42.4	48.3
Faster R-CNN*	ResNet-101-FPN	2x	39.7	61.4	43.3	22.3	42.9	50.4
Faster R-CNN*	ResNext-101-32x4d-FPN	1x	40.5	62.8	44.0	24.3	43.9	50.2
Faster R-CNN*	ResNext-101-64x4d-FPN	1x	41.7	64.1	45.4	25.0	45.1	52.1
Faster R-CNN*	MobileNet-v2-FPN	1x	32.6	54.6	34.3	18.6	34.5	41.0
Mask R-CNN*	ResNet-50-FPN	1x	37.5(34.4)	59.4(56.3)	40.6(36.6)	22.1(18.6)	40.6(37.2)	46.2(44.5)
Mask R-CNN*	ResNet-101-FPN	1x	39.8(36.3)	61.6(58.5)	43.3(38.7)	22.9(19.2)	43.2(39.3)	49.7(47.4)
Mask R-CNN*	ResNet-101-FPN	2x	40.8(36.9)	62.2(59.1)	44.6(39.6)	22.7(19.1)	44.0(39.9)	51.8(48.9)
RetinaNet (ours)	ResNet-50-AugFPN	1x	37.5[**+1.6**]	58.4	40.1	21.3	40.5	47.3
RetinaNet (ours)	MobileNet-v2-AugFPN	1x	34.0[**+1.3**]	54.0	36.0	18.6	36.0	44.0
FCOS (ours)	ResNet-50-AugFPN	1x	37.9[**+0.9**]	58.0	40.4	21.2	40.5	47.9
Faster R-CNN (ours)	ResNet-50-AugFPN	1x	38.8[**+2.3**]	61.5	42.0	23.3	42.1	47.7
Faster R-CNN (ours)	ResNet-101-AugFPN	1x	40.6[**+1.7**]	63.2	44.0	24.0	44.1	51.0
Faster R-CNN (ours)	ResNet-101-AugFPN	2x	41.5[**+1.8**]	63.9	45.1	23.8	44.7	52.8
Faster R-CNN (ours)	ResNext-101-32x4d-AugFPN	1x	41.9[**+1.4**]	64.4	45.6	25.2	45.4	52.6
Faster R-CNN (ours)	ResNext-101-64x4d-AugFPN	1x	43.0[**+1.3**]	65.6	46.9	26.2	46.5	53.9
Faster R-CNN (ours)	MobileNet-v2-AugFPN	1x	34.2[**+1.6**]	56.6	36.2	19.6	36.4	43.1
Mask R-CNN (ours)	ResNet-50-AugFPN	1x	39.5[**+2.0**](36.3[**+1.9**])	61.8(58.7)	42.9(38.8)	23.4(19.7)	42.7(39.2)	49.1(47.5)
Mask R-CNN (ours)	ResNet-101-AugFPN	1x	41.3[**+1.5**](37.8[**+1.5**])	63.5(60.4)	44.9(40.4)	24.2(20.4)	44.8(41.0)	52.0(49.8)
Mask R-CNN (ours)	ResNet-101-AugFPN	2x	42.4[**+1.6**](38.6[**+1.7**])	64.4(61.4)	46.3(41.4)	24.6(20.6)	45.7(41.6)	54.0(51.4)

Table 1. Comparison with the state-of-the-art methods on COCO test-dev. The symbol '*' means our re-implementation results. For Mask R-CNN, the results in ( ) means the corresponding mask results. The number in [] stands for the relative improvement. The training schedule follows the setting as Detectron [11].

3.3. Soft RoI Selection

In FPN, feature for each RoI is obtained by pooling on one certain feature level, which is chosen according to the scale of that RoI heuristically. Generally, small RoIs are assigned to features of lower levels while large RoIs are assigned to that of higher levels. Under this strategy, two RoIs with similar sizes may be assigned to different levels. This can produce sub-optimal results because it is ambiguous which feature level contains the most important information of an RoI. It is challenging to design a perfect strategy to allocate the RoIs.

PANet [25] addresses this by pooling RoI features from all levels and using the maximum of RoI features adapted by fully connected layers to refine the proposals. It improves the performance of instance segmentation but the extra fully connected layers increase the parameters significantly. Furthermore, the max operation only selects the feature points with the highest responses and ignores the features with lower responses in other levels that may be also beneficial for recognition. This may impedes the features at different levels from being fully exploited. Therefore, we propose Soft RoI Selection, which learns to generate better RoI features from features at all pyramid levels by parameterizing the procedure of RoI pooling. Soft RoI Selection introduces adaptive weights to better measure the importance of feature inside the RoI region at different levels. The final RoI features are generated based on the adaptive weightes instead of heuristic RoI assignment or max operation.

Specifically, we first pool features from all pyramid levels for each RoI. Then instead of adapting the RoI features with fully connected layers like PANet, we exploit an Adaptive Spatial Fusion module (ASF), which is also a component in Residual Feature Augmentation, to fuse these features adaptively. It generates different spatial weight maps for RoI features from different levels and the RoI features are fused with weighted aggregation. ASF only consists of two convolution layers and consumes much fewer parameters than the extra fully connected layers used in PANet. In this way, Soft RoI Selection parameterizes the procedure of RoI pooling. It can be leaned by back-propagation with other components in the network and does not rely on a heuristically designed strategy.

4. Experiments

4.1. Dataset and Evaluation Metrics

We perform all experiments on the MS COCO detection dataset with 80 categories. It contains 115k images

for training (*train*2017), 5k images for validation (*val*2017) and 20k images for testing (*testdev*). The labels of *testdev* are not released publicly. We train models on *train*2017 and report results of ablation study on *val*2017. The final results are reported on *testdev*. All reported results follow standard COCO-style Average Precision (AP) metrics.

4.2. Implementation Details

All experiments are implemented based on mmdetecton [2]. Following [20, 22, 37], the shorter sizes of input images are resized to 800 pixels. By default, we train the models with 8 GPUs (2 images per GPU) for 12 epochs. The initial learning rate is set as 0.02 and it decreases by a ratio of 0.1 after the 8th and 11th epoch respectively. λ in Equ. 1 is set as 0.25; as for the setting of ratio-invariant adaptive pooling, three alphas $\alpha_1, \alpha_2, \alpha_3$ with values as 0.1, 0.2, and 0.3 respectively are chosen if not noted specifically. All other hyper-parameters in this paper follow mmdetection.

4.3. Main Results

In this section, we evaluate AugFPN on COCO *test-dev* set and compare with other state-of-the-art one-stage and two-stage detectors. For a fair comparison, we re-implement the corresponding baseline methods equipped with FPN. All results are shown in Table 1. By replacing FPN with AugFPN, Faster R-CNN using ResNet50 as backbone (denoted as ResNet50-AugFPN) achieves 38.8 AP, which is 2.3 points higher than Faster R-CNN based on ResNet50-FPN. Besides, AugFPN can consistently bring non-negligible performance even with more powerful backbone networks. For example, when using ResNext101-32x4d and ResNext101-64x4d as the feature extractors, our method still improves the performance by 1.4 and 1.3 AP, respectively.

Obviously, Faster R-CNN with AugFPN significantly improves FPN when using powerful models like ResNet50 as backbone. Now we test whether AugFPN is suitable for light-weight models, *i.e.* MobileNet-V2 [35]. As shown in Table 1, Faster R-CNN with MobileNet-v2-AugFPN outperforms MobileNet-v2-FPN by 1.6 AP under $1\times$ learning rate schedule.

As for one-stage detectors, we validate the effectiveness of AugFPN on two different types of detectors, *i.e.* anchor-based RetinaNet [23] and anchor-free FCOS [37]. Since no concept of RoIs exist in these two detectors, Soft RoI Selection is not included in this case. Therefore, the outputs of detectors instead of RPN are used by the Consistent Supervision module in the training phase. As shown in Table 1, RetinaNet can be improved by 1.6 AP and 1.3 AP respectively when using ResNet50 or MobileNet-v2 as backbone. Meanwhile, FCOS is boosted to 37.9 AP from 37.0 AP when replacing FPN with AugFPN. The improvements show that the other two components still improve the feature representation of feature pyramid a lot even without including Soft RoI Selection.

CS	**RFA**	**SRS**	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
			36.3	58.3	39.0	21.4	40.3	46.6
✓			37.2	59.2	40.1	21.8	40.9	47.8
	✓		37.3	59.8	40.4	22.5	41.3	47.2
		✓	37.1	59.1	40.1	21.8	41.3	47.5
✓	✓		37.7	60	40.8	22.8	41.4	48.4
✓		✓	38.0	60.3	41.5	22.9	41.9	48.0
	✓	✓	37.9	60.3	40.7	23.6	41.8	47.9
✓	✓	✓	38.7	61.2	41.9	24.1	42.5	49.5

Table 2. Effect of each component. Results are reported on COCO *val*2017. **CS**: Consistent Supervision, **RFA**: Residual Feature Augmentation, **SRS**: Soft RoI Selection

Finally, we evaluate AugFPN on Mask R-CNN. By replacing FPN with AugFPN, Mask RCNN with ResNet50 is improved by 2.0 AP on the detection and 1.9 AP on instance segmentation. When using ResNet101 as backbone, the improvement of AugFPN reaches 1.5 AP on the detection and 1.5 AP on instance segmentation respectively. As can be seen in Table 1, AugFPN brings consistent improvements on various backbones, detectors and even different tasks. This verifies the robustness and generalization ability of AugFPN.

4.4. Ablation Study

In this section, we conduct extensive ablation experiments to analyze the effects of individual components in our proposed method.

Ablation studies on importance of each components. To analyze the importance of each component in AugFPN, Consistent Supervision, Residual Feature Augmentation and Soft RoI Selection are gradually applied to the model to validate the effectiveness. Meanwhile, the improvements brought by combination of different components are also presented to demonstrate that these components are complementary to each other. The baseline method for all ablation studies is Faster R-CNN with ResNet50-FPN. All results are shown in Table 2.

As shown in Table 2, Consistent Supervision improves the baseline method by 0.9 AP. This benefits from that Consistent Supervision narrow semantic gaps between the features after lateral connection and improves their semantic representation simultaneously. It is worthy to note that Consistent Supervision does not introduce extra parameters in inference. Therefore it is cheap to add it to any other FPN based detection models.

Residual Feature Augmentation improves the detection performance from 36.3 to 37.3 AP. It can be seen that results of objects in small, medium and large scale are all improved, which means the complementary information added to M_5 also benefits the feature maps at lower levels and im-

Setting	λ	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
no supervision	0.0	36.3	58.3	39.0	21.4	40.3	46.6
single level	1.2	36.7	58.5	39.7	21.3	40.1	47.3
single level	1.0	37.0	58.9	40.2	21.8	40.4	47.5
single level	0.5	36.9	58.7	40.0	21.7	40.9	47.4
single level	0.25	36.7	58.7	39.8	21.5	40.3	47.2
all level	0.5	36.9	58.8	39.9	21.8	40.7	47.1
all level	0.25	37.2	59.2	40.1	21.8	40.9	47.8
all level	0.125	37.1	58.9	40.1	22.3	40.9	47.4

Table 3. Ablation studies of Consistent Supervision on COCO *val*2017.

Fusion Type	Pooling Type	α	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
baseline		-	36.3	58.3	39.0	21.4	40.3	46.6
sum	GMP	-	34.5	56.6	36.8	21.9	38.3	42.4
sum	GAP	-	36.8	59.3	39.7	22.1	40.9	46.7
sum	RA-AP	0.1,0.2,0.3	37.1	59.8	39.9	22.7	41.1	47.3
ASF	RA-AP	0.1	37.1	59.6	40.2	22.3	40.9	47.2
ASF	RA-AP	0.1,0.2	37.2	59.4	40.1	22.4	41.1	47.7
ASF	RA-AP	0.1,0.2,0.3	37.3	59.8	40.4	22.5	41.1	47.4
ASF	RA-AP	0.1,0.2,0.3,0.4	37.4	59.9	40.5	22.5	41.1	47.9
ASF	RA-AP	0.1,0.2,0.4	37.3	59.7	40.2	22.9	41.3	47.2
ASF	RA-AP	0.1,0.2,0.5	37.2	59.7	40.3	22.2	41.1	47.0
ASF	PSP	-	37.0	59.5	40.1	22.8	40.9	47.3
PSP	PSP	-	36.9	59.5	39.6	22.3	40.9	46.8

Table 4. Ablation studies of Residual Feature Augmentation on COCO *val*2017. GMP, GAP means Global Max Pooling and Global Average Pooling respectively. RA-AP means ratio-invariant average pooling. ASF means Adaptive Spatial Fusion.

proves the feature representation of feature pyramid.

Soft RoI Selection brings 0.8 AP improvement to the baseline method. Specifically, the improvements of AP_m (+1.0 AP) and AP_l (+0.9 AP) contribute most to the final improvement. These results indicate that adaptive spatial fusion enables larger RoIs, which are originally assigned to higher feature levels, to incorporate features from lower levels that contain more information of spatial details.

When combining any two of three components, the improvement over the baseline method is much higher. For example, Consistent Supervision and Soft RoI Selection together can lead to 1.7 AP improvement. When three components are all integrated into the baseline method, it can achieve 38.7 AP with 2.4 AP improvement. These results indicate that these three components are complementary to each other and tackle different problems in FPN.

Ablation studies on Consistent Supervision. Experiment results related with three settings of Consistent Supervision are presented in Table 3. The first setting is the baseline method, where λ in Equ. 1 is set as zero. The second setting is *single level* supervision, which only applies supervision signals to the feature map that RoIs are assigned to according to the assignment strategy of RoIs in FPN [22]. The third setting is *all level* supervision, which enforces supervision signals to feature maps of all levels.

When using *single level* supervision, the baseline method can be improved by 0.7 AP by setting λ as 1.0. The improvement becomes smaller when λ is set as other values. By applying supervision signals on feature maps at all levels, *all level* supervision obtains better results than both *single level* setting and baseline method. It can be seen that when λ is set as 0.25, *all level* setting brings 0.5 and 0.9 AP improvement than *single level* setting and baseline model, respectively. The superiority of *all level* setting verifies that forcing feature maps at all levels to learn similar semantic information is an effective practice to narrow the semantic gap between them and improves the performance of the resulting feature pyramid.

Ablation studies on Residual Feature Augmentation. The results of ablation studies on Residual Feature Augmentation (RFA) are shown in Table 4. We first explore the influence of pooling type by using global pooing instead of ratio-invariant adaptive pooling. Since there is only one branch, Adaptive Spatial Fusion (ASF) is not adopted. Two types of global pooling, Global Max Pooling (GMP) and Global Average Pooling (GAP), are tested in the experiments. From the results shown in Table 4, we observe that GMP is inferior to GAP. GAP improves the baseline method by 0.5 AP while GMP degrades the accuracy instead, which means average pooling is more robust than max pooling because output of max pooling may be disturbed by the peak noises in feature maps greatly.

Based on this observation, we replace GAP with ratio-invariant adaptive average pooling (RA-AP). We firstly choose an α setting of three alphas with values as 0.1, 0.2 and 0.3 respectively. The influence of different α setting will be discussed afterward. For a fair comparison, the pooled context features are directly fused by summation instead of ASF. As shown in the fourth row in Table 4, RA-AP brings 0.8 AP and 0.3 AP improvement over the baseline method and GAP, which validates the effectiveness of diverse context brought by the residual branch. By combining ASF with RA-AP using the same α setting, the final result can be further boosted to 37.3 AP, which is 1.0 AP higher than the baseline method.

The influence of different α setting is also investigated. Although mAP increases as the number of α increases, as can be seen from Table 4, our final model adopts the setting of three alphas for a better trade-off between complexity and accuracy. In addition, we explore how different α values impact the performance and the experimental results are shown in the fourth part of table 4. When values of α are set as other values, the performance is even worse or shows no more improvement. To further validate the effectiveness of RFA, we conduct experiments where components of RFA are replaced with that of PSPNet[41] gradually. The superior results shows that RFA can preserve more information beneficial for recognition by not disturbing the original ratio of features and absorbing feature adaptively.

Ablation studies on Soft RoI Selection. We first study different methods of fusing RoI features. The first one is sum fusion and the second one is max fusion. The only difference between max fusion in this setting and Adaptive

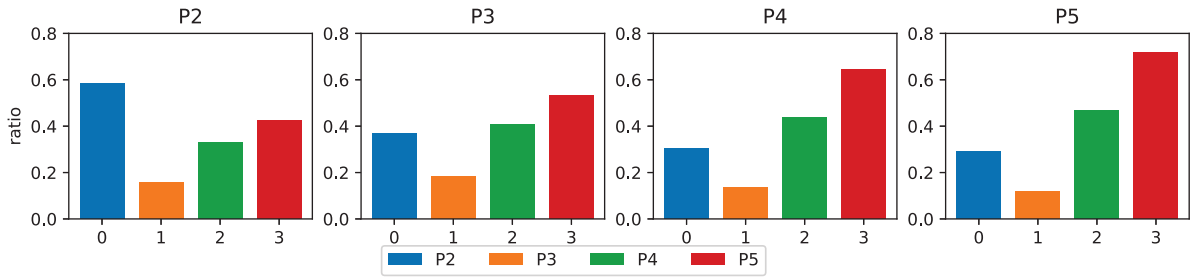


Figure 4. Ratios of features pooled from each pyramid level with Soft RoI Selection. The figures from left to right correspond to the RoIs originally assigned to $P_2 - P_5$. The results are obtained on COCO *val*2017.

Setting	Fusion type	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
baseline		36.3	58.3	39.0	21.4	40.3	46.6
SRS	sum	36.6	59.0	39.1	22.3	40.6	46.4
SRS	max	36.5	58.5	39.2	21.6	40.2	46.9
SRS	ACF	37.0	59.2	39.8	22.0	41.2	46.8
SRS	ASF	37.1	59.1	40.1	21.8	41.3	47.5
PANet	max	36.9	59.0	39.7	21.6	40.7	47.5

Table 5. Ablation studies of Soft RoIs Selection on COCO *val*2017. SRS, ACF and ASF is the acronym of Soft RoI Selection, Adaptive Channel Fusion and Adaptive Spatial Fusion respectively.

Pooling in PANet [25] is that we do not introduce extra fully connected layers to adapt the RoI features because it would significantly increases the parameters. The third one is the Adaptive Channel Fusion (ACF) as shown in Figure 3(b). It is inspired by the SE module [15] but with a different goal of fusing different RoI features from the perspective of channel importance. The fourth one is the Adaptive Spatial Fusion (ASF) module as shown in Figure 3(a). Experimental results on these methods are shown in Table 5.

From the results we can observe that sum fusion and max fusion improve baseline method by 0.3 and 0.2 AP respectively. By using ACF to fuse RoI features adaptively, the baseline method obtains 0.7 AP improvement. When ACF is replaced with ASF, which is the setting of Soft RoI Selection, the final model achieves 37.1 AP and outperforms the baseline method by 0.8 AP. These results indicate that by enabling the procedure of RoI feature selection to learn with other components, Soft RoI Selection can produce more powerful representations of RoIs. In addition, we also implement Adaptive Pooling in PANet and it achieves worse result while consuming much more parameters than ours (38.53M vs 0.27M).

In order to analyze the ratios of features at different levels absorbed by ASF, we divide RoI proposals on *val*2017 into four levels according to the levels they are originally assigned to. For each RoI, we average over all positions on each weight map generated by ASF and obtain four ratios corresponding to four feature levels. Finally, for all RoIs that belong to one certain level, four ratio values are separately averaged over these RoIs. The results corresponding to four pyramid levels are illustrated in Figure 4. Obviously, features from all levels contribute together to generate better RoI features, which indicates that features from all levels are beneficial for the recognition of each RoI. It can be seen that the RoIs originally assigned to level P2 still requires more semantic information from P5 beside the information propagated from higher levels. Meanwhile, the RoIs originally assigned to P3-5 all requires much detailed appearance information from P2, which may be lost due to down-sampling.

4.5. Runtime Analysis

We also measure the time of training and testing when FPN is replaced with AugFPN. Specifically, the training time of Faster-RCNN with ResNet50-AugFPN is about 1.1 hour and that of Faster-RCNN with ResNet50-FPN is nearly 0.9 hour for each epoch on COCO dataset with the same batch size of 16. As for the inference time, AugFPN can run at 11.1 fps and FPN can run at 13.4 fps for images with a shorter size of 800 pixels. The inference time is the average inference time over COCO *val5000* split including the time of data loading, network forwarding, and post-processing. All the runtimes are tested on Tesla V100.

5. Conclusion

In this paper, we analyze the inherent problems along with FPN and find that the multi-scale features are not fully exploited. Based on this observation, we propose a new feature pyramid network named AugFPN to further exploit the potential of multi-scale features. By integrating three simple yet effective components, *i.e.* Consistent Supervision, Residual Feature Augmentation, and Soft RoI Selection, AugFPN can improve the baseline method by a large margin on the challenging MS COCO dataset.

Acknowledgement

This work was supported by the Major Project for New Generation of AI under Grant No. 2018AAA0100400, the National Natural Science Foundation of China under Grants 61876180, 61573352, 91646207, 61976208, and 61773377, the Young Elite Scientists Sponsorship Program by CAST (2018QNRC001), and the Beijing Natural Science Foundation under Grant 4162064.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 2

[2] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. mmdetection. https://github.com/open-mmlab/mmdetection, 2018. 6

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *T-PAMI*, 40(4):834–848, 2018. 3

[4] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. 2, 5

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2

[6] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *CoRR*, abs/1701.06659, 2017. 2

[7] Golnaz Ghiasi, Tsung-Yi Lin, Ruoming Pang, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. 2019. 2

[8] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *ICCV*, 2015. 3

[9] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 1, 2

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2

[11] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron, 2018. 5

[12] Chaoxu Guo, Bin Fan, Jie Gu, Qian Zhang, Shiming Xiang, Veronique Prinet, and Chunhong Pan. Progressive sparse local attention for video object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 2, 3, 5

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 2

[15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 8

[16] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q Weinberger. Multi-scale dense networks for resource efficient image classification. 2017. 2

[17] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. *CoRR*, abs/1811.11721, 2018. 3

[18] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018. 2

[19] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *AISTATS*, 2015. 2

[20] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. *CoRR*, abs/1901.01892, 2019. 2, 6

[21] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Light-head r-cnn: In defense of two-stage object detector. *CoRR*, abs/1711.07264, 2017. 2

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1, 2, 3, 5, 6, 7

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1, 2, 5, 6

[24] Songtao Liu, Di Huang, et al. Receptive field block net for accurate and fast object detection. In *ECCV*, 2018. 2

[25] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 2, 5, 8

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 1, 2, 5

[27] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *CoRR*, abs/1506.04579, 2015. 3

[28] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid r-cnn. In *CVPR*, 2019. 5

[29] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *CVPR*, 2019. 2, 5

[30] Zheng Qin, Zeming Li, Zhaoning Zhang, Yiping Bao, Gang Yu, Yuxing Peng, and Jian Sun. Thundernet: Towards real-time generic object detection. *CoRR*, abs/1903.11752, 2019. 2, 3

[31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1, 2

[32] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017. 2, 5

[33] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 2

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 2

[35] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 6

[36] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In *CVPR*, 2018. 2

[37] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019. 6

[38] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 2013. 2

[39] Xingyu Zeng, Wanli Ouyang, Junjie Yan, Hongsheng Li, Tong Xiao, Kun Wang, Yu Liu, Yucong Zhou, Bin Yang, Zhe Wang, et al. Crafting gbd-net for object detection. *T-PAMI*, 2018. 3

[40] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *CVPR*, 2018. 2

[41] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2, 3, 4, 7

[42] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. *CoRR*, abs/1901.08043, 2019. 2

[43] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019. 2