2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

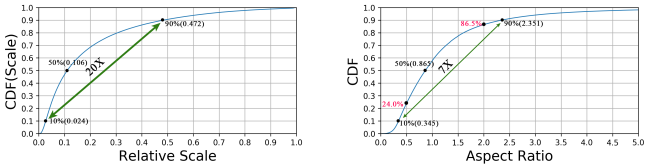
Bidirectional Matrix Feature Pyramid Network for Object Detection

Wei Xu, Yi Gan, Jianbo Su
Research Center of Intelligent Robotics
Shanghai Jiao Tong University
Shanghai, China
Email: {xuweiVG, 627780002ganyi, jbsu}@sjtu.edu.cn

***Abstract*—Feature pyramids are widely used to improve scale invariance for object detection. Most methods just map the objects to feature maps with relevant square receptive fields, but rarely pay attention to the aspect ratio variation, which is also an important property of object instances. It will lead to a poor match between rectangular objects and assigned features with square receptive fields, thus preventing from accurate recognition and location. Besides, the information propagation among feature layers is sparse, namely, each feature in the pyramid may mainly or only contain single-level information, which is not representative enough for classification and localization sub-tasks. In this paper, Bidirectional Matrix Feature Pyramid Network (BMFPN) is proposed to address these issues. It consists of three modules: Diagonal Layer Generation Module (DLGM), Top-down Module (TDM) and Bottom-up Module (BUM). First, multi-level features extracted by backbone are fed into DLGM to produce the base features. Then these base features are utilized to construct the final feature pyramid through TDM and BUM in series. The receptive fields of the designed feature layers in BMFPN have various scales and aspect ratios. Objects can be correctly assigned to appropriate and representative feature maps with relevant receptive fields depending on its scale and aspect ratio properties. Moreover, TDM and BUM form bidirectional and reticular information flow, which effectively fuses multi-level information in top-down and bottom-up manner respectively. To evaluate the effectiveness of our proposed architecture, an end-to-end anchor-free detector is designed and trained by integrating BMFPN into FCOS. And the center-ness branch in FCOS is modified with our Gaussian center-ness branch (GCB), which brings another slight improvement. Without bells and whistles, our method gains +3.3%, +2.4% and +2.6% AP on MS COCO dataset from baselines with ResNet-50, ResNet-101 and ResNeXt-101 backbones, respectively.**

I. Introduction

In recent years, significant progress has been made in object detection due to the advance of deep neural networks [1]–[3] and well-annotated datasets [4], [5]. Most state-of-the-art anchor-based methods [6]–[8] rely on anchor boxes of various scales and aspect ratios as candidates to search potential regions of interest. These approaches always require substantial anchor boxes as well as the scales and aspect ratios of anchors to be carefully designed, which results in redundant negative samples and additional hyper-parameters. In order to address these drawbacks of anchor-based approaches, many anchor-free approaches [9]–[14] have been proposed and achieved excellent performance in recent years. These anchor-free detectors can be further divided into two types according to whether it is based on keypoint or not. CornerNet [13] is a typical keypoint based method, which utilizes keypoint estimation to directly detect the top-left and bottom-right corner points. And then Associative Embedding [15] method is applied to group the corners belonging to the same object. As for non-keypoint based detection methods like FCOS [9], all locations in the features are viewed as training samples and the bounding boxes of each location are regressed directly.



(a) Fraction of bboxes *vs* scale of bboxes relative to the image

(b) Fraction of bboxes *vs* aspect ratio of bboxes

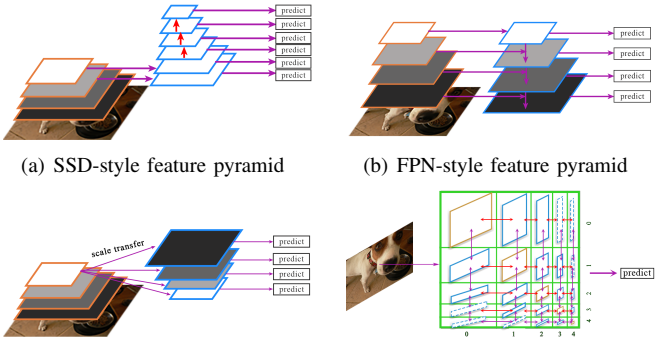
Fig. 1. Statistical analysis of annotated bounding boxes (bboxes) in COCO. CDF is the abbreviation of cumulative distribution function.

Despite the significant differences in the detection approaches, detecting objects at different scales is a critical problem. Large scale variation across object instances can be seen clearly in Fig. 1(a). The scale relative to the image of the smallest and largest 10% is 0.024 and 0.472 in COCO respectively, which results in scale variations of almost 20 times [16]. To address the problems arising from scale variation, various feature pyramids are designed and widely used by mapping object instances to layers with relevant receptive fields. For example, as illustrated in Fig. 2, SSD [7] directly constructs the feature pyramid with two layers from the backbone and the other four layers obtained by stride 2 convolutions. FPN [17] combines the deep and shallow layers to construct the high-level semantic feature pyramids in a top-down manner. STDN [18] uses DenseNet [19] generate more powerful features and then final feature pyramid is produced by pooling and scale-transfer operations.

However, can the assigned feature layer fit perfectly with the object only considering the scale variation? The aspect ratio variation is also analysed and shown in Fig. 1(b). The aspect ratio of nearly 40% of objects are distributed outside the range of (0.5, 2), which means that there are quite a few slender and stubby objects. Besides, the aspect ratio of

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412229



(a) SSD-style feature pyramid (b) FPN-style feature pyramid

(c) STDN-style feature pyramid (d) Our bidirectional matrix feature pyramid

Fig. 2. Illustrations of four kinds of feature pyramids. Bidirectional arrows represent bidirectional information flow in our bidirectional matrix feature pyramid.

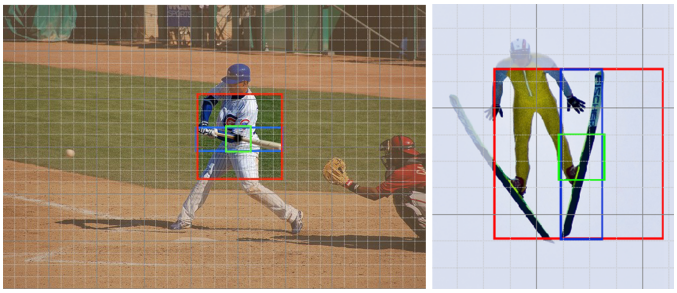


Fig. 3. Examples of objects assignment. The regions in the red and green rectangles are the receptive fields of feature layers at different resolutions respectively, when mapping the objects in the blue rectangle. It leads to a poor match between objects and assigned feature layers.

the lowest and highest 10% of object instances is 0.345 and 2.351 in COCO respectively, which results in aspect ratio variation of almost 7 times. This reminds us that the aspect ratio variation should also be taken into account. However, the receptive filed of feature maps in the above-mentioned feature pyramids are designed to be square, which leads to a poor match between rectangular objects and assigned feature maps, thus preventing from accurate recognition and location. For a vivid explanation, as shown in Fig. 3, when an object like a baseball bat or a ski inside the blue rectangle is assigned to a layer whose receptive field is in the red square region, this region will contain a large amount of disturbing and useless information. It is detrimental for accurate classification and localization sub-tasks. Moreover, when the region in green square is the relative receptive field which can not cover the object, the loss of information makes it hard to identify what the object is, let alone pinpoint it. Therefore, the poor match between objects and assigned features is bound to occur among a rectangular and a square receptive filed. It is even deadly for some objects of extreme high or low aspect ratios.

Besides, each feature in the pyramid is always required to be representative enough and contain rich information for higher detection performance. In general, the shallow low-level features are more content descriptive while the deep high-level features carry more semantic information. And the high-level and low-level information are more discriminative and helpful for classification and localization sub-tasks respectively. It is less than reliable to make predictions directly from different feature layers of the backbone network for consistent predictions. The typical solution is to integrate the feature maps at different resolutions before predictions. But the information propagation among feature layers is sparse in aforementioned approaches. For example, FPN [17] only forms a single and simple top-down pathway to propagate high-level information. It will make the integrated features focus more on adjacent resolution but less on others. In other words, Each feature in the pyramid may mainly or only contain single-level information, which is not rich enough for classification and localization sub-tasks, thus limiting the detection performance.

In this paper, Bidirectional matrix feature pyramid network (BMFPN) is proposed to address the issues mentioned above. It consists of three modules: Diagonal Layer Generation Module (DLGM), Top-down Module (TDM) and Bottom-up Module (BUM). First, multi-level features extracted by backbone are fed into DLGM to produce base features. Then the base features are utilized to construct the final feature pyramids through TDM and BUM in series. The receptive fields of the designed feature layers in BMFPN have various scales and aspect ratios. Objects can be correctly assigned to more appropriate and representative feature layers with relevant receptive fields depending on its scale and aspect ratio properties. Moreover, TDM and BUM forms bidirectional and reticular information flows, which effectively fuses multi-level information in top-down and bottom-up manner respectively. To evaluate the effectiveness of our proposed architecture, an end-to-end anchor-free detector is designed and trained by integrating BMFPN into FCOS [9]. And the center-ness branch in FCOS is modified with our Gaussian center-ness branch (GCB), which brings another slight improvement. The overall pipeline of our model is shown in Fig. 4.

The rest of this paper is organized as follows: some related works are discussed in Section II. Our proposed method is presented in Section III and experimental results is reported in Section IV. Finally, all our work is concluded in Section V.

II. Related Works

Detecting objects at different scales is a critical problem in object detection. The solutions to address it can be roughly divided into methods predicting from the hierarchy of the backbone features, methods based on feature pyramids, methods based on image pyramids and methods combining image and feature pyramids [20]. Feature pyramids are widely used to improve scale invariance owing to its fast inference speed and low memory consumption. Various methods have been proposed to enhance the feature representations for high detection performance.

Feature pyramid network (FPN) [17] combines multi-level features of the backbone to construct the high-level semantic

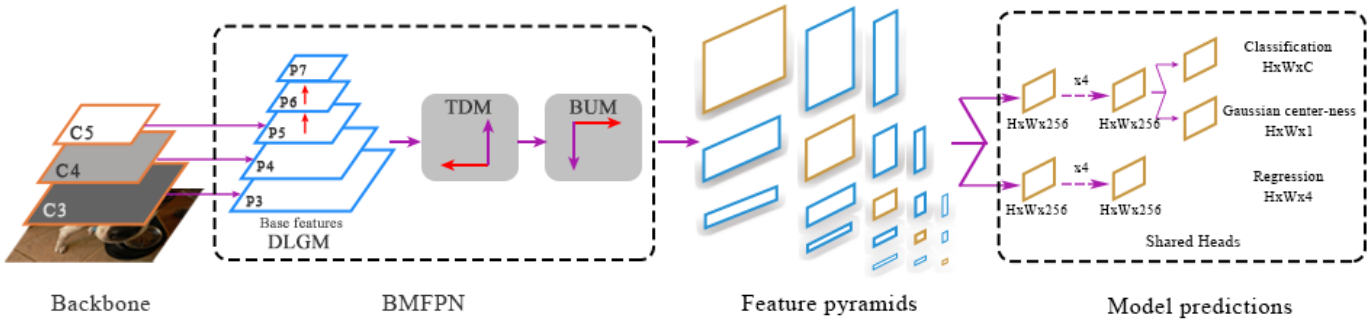


Fig. 4. The overall pipeline of our model. DLGM utilizes multi-level features extracted by backbone to generate the base features. Then the base features are fed into TDM and BUM in series to construct feature pyramids for final model predictions.

feature pyramid in a top-down manner. However, the deeper feature layer has difficulty in accessing accurate localization information because of the long bottom-up pathway with aggressive down-sampling. Path Aggregation Network (PANet) [21] further enhances the deep layers with accurate localization information in lower layers by means of bottom-up path augmentation. Different from these sequential enhancement methods, Libra R-CNN [22] first rescales and integrates the FPN layers equally to obtain a balanced semantic feature map, which is then rescaled using a reverse procedure to strengthen the original features. And non-local neural networks [23] are used to further enhance the integrated features. Multi-Level Feature Pyramid Network (MLFPN) [24] extracts multi-level and multi-scale features and uses an adaptive attention mechanism to aggregate the features into the multi-level feature pyramids. AugFPN [25] analyzes the design defects of FPN and proposes consistent supervision, residual feature augmentation and soft RoI selection modules to address these problems. Rather than hand-crafted architectures, Neural Architecture Search FPN (NAS-FPN) [26] aims to learn a better feature pyramid architecture for object detection with the help of neural architecture search methods.

Matrix Nets(xNets) first [27] provides a feasible solution to detect objects of different aspect ratios. It generates several matrix layers and each layer is used to handle an object of specific size and aspect ratio. Compared with xNets, our BMFPN applies transposed convolutions(deconvolutions) and dilated convolutions [28] with asymmetric strides to generate a matrix-like feature pyramid. As the effective receptive field only occupies a fraction of the theoretical receptive field [29], dilated convolutions can increase the size of effective receptive filed. Besides, the dilated convolution is more conducive to covering the wholes rectangular objects to get global information. Moreover, our TDM and BUM form bidirectional and reticular information flows, which is what xNets lacks. Each layer in our BMFPN contains multi-level information, which is representative for higher detection performance.

III. Bidirectional Matrix Feature Pyramid Network

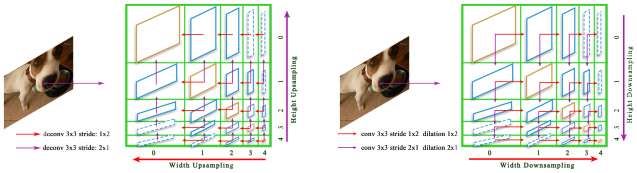
In this section, we instantiate our BMFPN by integrating it into the anchor-free detector FCOS [9]. Besides, the center-ness branch in FCOS is modified with our Gaussian center-ness branch. The overall pipeline of our model is shown in Fig. 4. It consists of three modules: Diagonal Layer Generation Module, Top-down Module and Bottom-up Module. And we demonstrate our design from the following subsections: (1) Introduction of our baseline FCOS (III-A); (2) Diagonal Layer Generation Module (III-B); (3) Top-down Module (III-C); (4) Bottom-up Module (III-D); (5) Object assignment strategy (III-E); (6) 2D-Gaussian center-ness branch (III-F).

A. Fully Convolutional One-Stage Object Detection

FCOS [9] is an efficient anchor-free detector which avoids the complicated IoU computation and additional hyper-parameters by removing anchor boxes. And FPN [17] is applied to construct feature pyramids for model predictions. Its network architecture is based on RetinaNet [8]. Besides, it adds a center-ness branch paralleled with the classification branch to predict the "center-ness" of a location. It views locations in each feature map as training samples and regresses the bounding boxes directly. Each location falling into any ground-truth bounding box is considered as a positive sample while the others are negative. When assigning the samples to different feature levels, it directly sets the scale range of the object assigned to each feature layer. More specifically, it firstly compute the regression targets l^*, t^*, r^*, b^* (the distances to four boundaries of ground-truth bounding boxes: left, top, right, bottom) for each positive location on all feature levels. And then each sample is assigned to relative feature level according to $max(l^*, t^*, r^*, b^*)$. Focal loss [8] and IoU loss [30] are applied for classification and localization subtasks in training process. During testing, the final score fed into non-maximum suppression (NMS) process is computed by multiplying the corresponding classification score with the predicted center-ness map to suppress the low-quality detected bounding boxes.

B. Diagonal Layer Generation Module

In BMFPN, DLGM utilizes multi-level features extracted by backbone to produce the base features. As shown in Fig.4, it uses 1×1 and 3×3 convolution layers to compress the channels of C3 $\sim$ C5 to 256. And another two 3×3 convolution layers with stride 2 are applied to generate P6 and P7. P3 $\sim$ P7 denote



(a) Top-down Module (b) Bottom-up Module

Fig. 5. TDM and BUM. (a) Top-down Module. The base features produced by DLGM are viewed as the diagonal layers in the matrix. We fill in the rest of the matrix by upsampling these layers. (b) Bottom-up Module. It constructs multiple bottom-up pathways in a similar but reverse way compared with TDM.

the base features and their relevant receptive fields are square. Compared with FPN [17], it solely removes the upsampling operations without adding any parameters.

C. Top-down Module

TDM aims to enhance the high-level semantic information of feature maps. As shown in Fig.5(a), it forms a 5×5 matrix structure so these features are called matrix layers. And each entry i, j in the matrix represents a layer $L_{(i,j)}$. The diagonal layers are the base features produced by DLGM. The red arrows represent a shared 3×3 deconvolution with stride 1×2 while the purple arrows represent a shared 3×3 deconvolution with stride 1×2. There is an element-wise addition operation if a layer is pointed by several arrows. The layer $L_{(i,j)}$ is generated with width upsampling of 2^{4-i} and height upsampling of 2^{4-j} relative to the bottom right layer $L_{(4,4)}$. Every step to the left doubles the width of feature, while every step top doubles the height. For example, $Width_{L_{(1,0)}} = Width_{L_{(1,1)}}, Height_{L_{(1,0)}} = 2Height_{L_{(1,1)}}$. Different from FPN, TDM builds multiple top-down pathways to propagate high-level information from deep layers to shallow layers. It forms a reticular and dense information flow to construct the high-level semantic feature pyramids. It is worth mentioning that only the 5 diagonal layers in TDM are fed into BUM for computation reduction. And any one of these five layers can gather the information propagated from its deeper layers following the arrow in Fig.5(a), which makes the features more representative.

D. Bottom-up Module

BUM aims to construct a feature pyramid whose receptive field has different scales and aspect ratios. Compared with TDM, it builds multiple bottom-up pathways in a similar but reverse way to enhance features with accurate localization information existing in low-levels. As shown in Fig.5(b), the red arrows represent a shared 3×3 dilated convolution with stride 1×2 while the purple arrows represent a shared 3×3 convolution with stride 2×1. The layer $L_{(i,j)}$ is enhanced with width downsampling of 2^i and height downsampling of 2^j relative to the top left layer $L_{(0,0)}$. Each red arrow cuts the width of feature by half, while each purple arrow cuts the height by half. The receptive field of the diagonal layer is a square region, namely, the value of its aspect ratio of receptive field (RFAR) is 1. Owing to the asymmetric strides in convolutions, the receptive fields of non-diagonal feature layers change to a rectangular region. So our BMFPN can be adapted to objects with various scales and aspect ratios. Besides, the RFAR will be proportional between each matrix layer in theory. For example, $RFAR_{L_{(0,0)}} = 1, RFAR_{L_{(1,0)}} = 2, RFAR_{L_{(2,0)}} = 4$. $RFAR_{L_{(i,j)}}$ are the aspect ratio of receptive field of layer $L_{(i,j)}$.

Benefiting from this architecture, objects can be correctly assigned to appropriate feature maps with relevant receptive fields depending on its scale and aspect ratio properties. Those whose aspect ratios are very high or low will be assigned to the dashed layers, which are close to the top right or bottom left corners. Such objects are very rare, so these dashed layers like are $L_{(4,0)}$ only used to propagate information without making further model predictions. Besides, a shared layer (dilation convolution + RELU) between TDM and BUM is used to increase the size of effective receptive field and the non-linear ability of the model. Finally, TDM and BUM construct an effective feature pyramids whose receptive fields have various scales and aspect ratios. And these two modules form bidirectional and reticular information flow, which effectively fuses multi-level information in top-down and bottom-up manner respectively. It only introduces negligible parameters because of the shared deconvolution and dilated convolution layers. Moreover, these two modules are very flexible in connection mode. Any one of them can be the first in series. And they can also be connected in parallel. We will discuss it in IV-C.

E. Object assignment strategy

Different from FCOS, we set both the width and height ranges ($WR_{L_{(i,j)}}$ and $HR_{L_{(i,j)}}$) of object instances assigned to each matrix layer. Given the regression targets l^*, t^*, r^*, b^* (the distances to four boundaries of ground-truth bounding boxes) for a location, if a location satisfies $max(l^*, r^*) \in WR_{L_{(i,j)}}$ and $max(t^*, b^*) \in HR_{L_{(i,j)}}$, it is assigned to layer $L_{(i,j)}$ for further predictions. $WR_{L_{(i,j)}}$ and $HR_{L_{(i,j)}}$ are the width and height range of layer $L_{(i,j)}$ respectively.

And the width and height ranges are required to reflect the receptive field of each matrix layer. As we analyzed above, the aspect ratio of receptive field are proportional between each matrix layer in theory. So once the range of any layer is defined, the ranges of rest layers can be determined naturally. For example, if the range for layer $L_{(0,0)}$ is height ∈ (16, 32) and width ∈ (16, 32), the range for layer $L_{(1,0)}$ will be height ∈ (16, 32) and width ∈ (32, 64). However, a slight change in object size around the boundaries of two adjacent layers will make a different distribution results. We just add two factors α_1,α_2 by multiplying the lower and upper bound of the range respectively. The values of α_1 and α_2 are set to 0.9, 1.2 in all our experiments respectively. Besides, we choose (16, 32) as the base range of layer $L_{(0,0)}$. The detection performance can be further improved with careful parameter adjustment.

F. Gaussian Center-ness Branch

Given the regression targets l^*, t^*, r^*, b^* (the distances to four boundaries of ground-truth bounding boxes) for a loca-

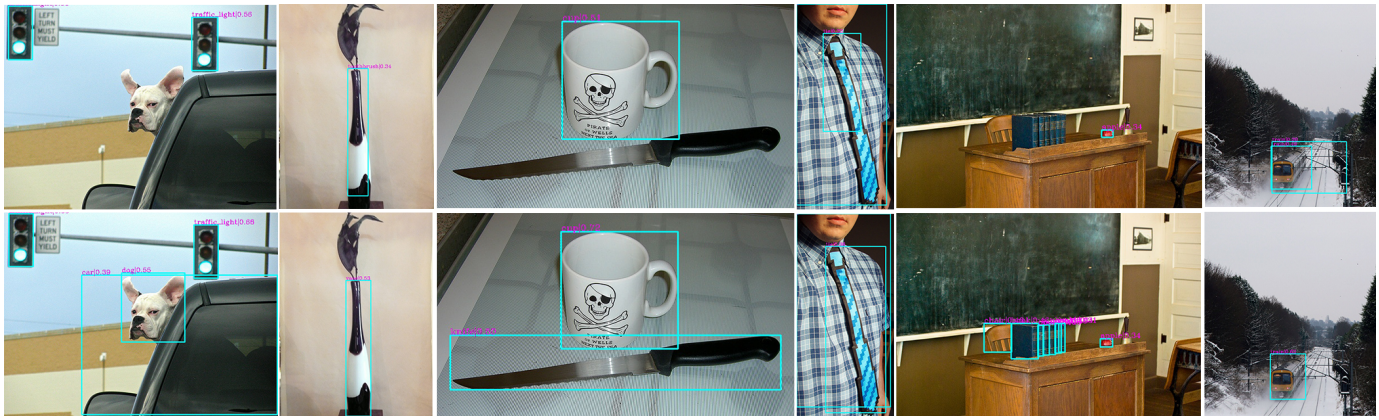


Fig. 6. Some comparison examples between FCOS (top) and our detector with BMFPN (bottom). Both are using ResNet-50 as backbone. Our BMFPN helps finding more challenging objects.

tion, FCOS [9] defines the center-ness target as,

$$centerness^* = \sqrt{\frac{\min(l^*, r^*)}{\max(l^*, r^*)} \times \frac{\min(t^*, b^*)}{\max(t^*, b^*)}} \ . \quad (1)$$

The center-ness ranges from 0 to 1 and it is trained with binary cross entropy (BCE) loss. When testing, the predicted center-ness map is utilized to get the final score map by an element-wise product operation with the classification score. As a result, these low-quality bounding boxes far from the centre might be filtered out by the common non-maximum suppression (NMS) process, which bringing significant improvement. However, it is mentioned in FCOS that using the ground-truth center-ness significantly improves AP to 42.1% during inference, meaning that there is much room for further improving the performance. An intuitive improvement is applying a new center-ness target. We propose Gaussian Center-ness Branch (GCB) which takes into account the aspect ratio of bounding boxes. Suppose (i, j) is inside the assigned feature map F_m of m-th annotated box, (x_0, y_0) is the corresponding central point, the Gaussian center-ness target is defined as,

$$Gauss\_centerness^* = \sqrt{e^{-\frac{(x_i - x_0)^2}{2\sigma_x^2} - \frac{(y_i - y_0)^2}{2\sigma_y^2}}}, \quad (2)$$

$$\sigma_x = \frac{w}{6\sigma} \ , \sigma_y = \frac{h}{6\sigma} \ , \quad (3)$$

$$w = l^* + r^*, h = t^* + b^*, \quad (4)$$

$$x_i - x_0 = \frac{|l^* - r^*|}{2}, y_i - y_0 = \frac{|t^* - b^*|}{2} \ , \quad (5)$$

The target is decided by the parameter σ, central location (x_0, y_0) and box size (h, w). Besides, as Eq. 4 and Eq. 5 show, h, w are the distance to central point along the x-axis and y-axis , which need not to be recalculated once the regression targets l^*, t^*, r^*, b^* are given. GCB ranges from 0 to 1 and the farther away from the centre, the lower the value. Our proposed GCB exhibits three advantageous properties: (1) We take into consideration the aspect ratio of the box in our Gaussian kernel, rather than the Gaussian kernel ($e^{-\frac{x^2+y^2}{2\sigma^2}}$) in CornerNet [13]. (2) GCB provides a flexible way by controlling σ to make the "energy" more concentrated in the central region of the bounding box, which is conducive to the suppression of low-quality samples far from the central points. We set $\sigma = 0.97$ in all our experiments. (3) As a bonus, our Gaussian center-ness loss is able to automatically keep the balance between classification loss and regression loss. We need not to adjust the weight of these tasks.

TABLE I
THE PERFORMANCE FROM THE BASELINE GRADUALLY TO ALL COMPONENTS INCORPORATED ON MS COCO VAL-2017 SPLIT.

GCB	DLGM	TDM	BUM	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
				36.7	55.8	39.2	21.0	40.7	48.4
✓				37.0	55.9	39.5	21.3	40.9	48.6
	✓			33.2	49.9	35.4	14.5	36.1	47.8
	✓	✓		37.6	56.7	40.3	21.0	41.5	49.6
	✓		✓	36.5	53.0	39.3	16.9	40.6	52.2
	✓	✓	✓	39.7	57.7	42.8	22.4	44.4	53.0
✓	✓	✓	✓	**40.0**	**57.7**	**43.1**	**22.8**	**44.5**	**53.4**

IV. EXPERIMENTS

All our experiments are conducted on the challenging MS COCO [5] dataset. Models are trained on the train2017 split containing 118k images and evaluated on val-2017 split including 5k images for all ablation studies. We report the final results on test-dev split (20k images) when comparing to state-of-the-art detectors. All reported results follow the standard COCO-style Average Precision (AP) metrics.

And our experiments are implemented based on MMDetection [31]. We use 4 GPUs with a total of 16 images per minibatch (4 images per GPU) for ResNet50 and ResNet101 models, and 8 GPUs with 2 images per GPU for ResNeXt101 models. Unless otherwise specified, the models are trained for 12 epochs with stochastic gradient descent (SGD). The initial learning rate is 0.01 and it is reduced by a factor of 10 after 8 and 11 epochs respectively. And all the ablation studies use ResNet50 as the backbone network. All the other hyper-parameters follow the default settings in MMDetection.

TABLE II
DETECTION RESULTS OF SOME CATEGORIES ON MS COCO VAL-2017 SPLIT. THE NUMBERS IN PARENTHESIS STANDS FOR THE RELATIVE AP IMPROVEMENT.

Method	airplane	snowboard	surfboard	fork	keyboard	toaster	scissors	toothbrush	refrigerator	tennis racket
FCOS	61.6	21.4	26.0	23.1	42.8	21.7	21.3	14.3	49.0	43.2
ours	69.5(**+7.9**)	32.3(**+10.9**)	34.7(**+8.7**)	39.7(**+6.6**)	50.8(**+8.0**)	36.4(**+14.7**)	32.0(**+10.7**)	22.5(**+8.2**)	55.1(**+6.1**)	49.3(**+6.1**)

TABLE III
COMPARISON BETWEEN DIFFERENT FEATURE PYRAMID STRUCTURES BASED ON FCOS.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params (M)
Baseline(FPN) [17]	36.7	55.8	39.2	21.0	40.7	48.4	32.02
PAFPN [21]	37.0	56.1	39.4	20.8	40.8	48.4	34.38
xNets [27]	37.9	56.0	40.7	21.1	43.0	50.9	33.20
BMFPN*	38.0	56.3	40.5	21.2	41.7	51.3	34.97
BMFPN	**39.7**	**57.7**	**42.8**	**22.4**	**44.4**	**53.0**	34.97

TABLE IV
COMPARISON BETWEEN DIFFERENT CONNECTION MODE ON MS COCO VAL-2017 SPLIT. TDM+BUM : PARALLEL MODE; BUM→TDM : BUM FIRST UNDER SERIES MODE; TDM→BUM : TDM FIRST UNDER SERIES MODE.

Mode	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
TDM+BUM	38.9	57.5	41.6	22.2	43.2	51.8
BUM→TDM	39.2	57.6	42.1	22.0	43.5	51.9
TDM→BUM	**39.7**	**57.7**	**42.8**	**22.4**	**44.4**	**53.0**

TABLE V
COMPARISON WITH DIFFERENT CONNECTION LAYERS BETWEEN TDM AND BUM ON MS COCO VAL-2017 SPLIT. MODE NO DENOTES TDM AND BUM ARE DIRECTLY CONNECTED WITHOUT ANY OTHER CONNECTION LAYER.

Mode	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
NO	39.4	57.7	42.1	22.3	43.6	53.3
Convolution	39.4	57.6	42.2	21.5	44.2	52.9
Dilated convolution	**39.7**	**57.7**	**42.8**	**22.4**	**44.1**	**53.0**

A. Component Ablation Studies

To analyze the importance of each proposed component, our ablation study reports the performance from the baseline gradually to all components incorporated. The results are shown in Table I. The first row is the results of FCOS implemented in MMDetection, which achieves 36.7% AP. And we only use its submitted version as our baseline without any improvements mentioned in its subsequent version, such as central sampling and GIoU [32].

1) Gaussian Center-ness Branch. When we directly replace the original center-ness branch with our Gaussian Center-ness Branch (GCB), it yields a 0.3 improvement on AP. And all the AP metrics are slightly improved.

2) Diagonal Layer Generation Module. DLGM is applied to generate the base features. When these features are directly used for detection, the AP decreases to 33.2%. It demonstrates that it is less than reliable to make predictions directly from the feature layers with single-level information.

3) Top-down Module. As shown in Table I, it comes to AP of 37.6% when we add TDM after DLGM, which brings a large margin. The improvement on object instances with small scale is most significant (AP_S increases 6.5%), which owes to the high-level semantic information from deeper feature layers. And it also improve the baseline by 0.9% AP.

4) Bottom-up Module. When just adding BUM after DLGM, AP has improved from 33.2% to 36.5% as illustrated in the fifth row in Table I. Especially, the AP_M and AP_L scores achieve 4.5%, 4.4% improvements respectively, owing to multiple bottom-up pathways to propagate the low-level localization information to these deeper layers. When combining TDM and BUM modules together, the performance of AP is improved by 2.1% and 3.2% respectively compared with each single module, which further indicates that the low-level and high-level information are complementary.

With all these components added to FCOS, improvement on AP is 3.3% over baselines. And the results shows that large size instances contribute most(+5.0%). Moreover, it makes more accurate detection with 3.9% improvement on AP_{75}. The improvement of small objects is not significant because of the existence of FPN in the baseline. Specifically, to find out what kinds of objects the BMFPN can detect, we show some detection results of the head to head comparison between the baseline and our detector with BMFPN in Fig. 6. Clearly, our model is better at finding challenging instances, such as very thin and obscured objects. And our method is more conducive to accurate location. Besides, we list the AP of some categories for an intuitive comparison. As shown in Table II, the detection results of these slender objects in daily life are greatly improved. It further verifies that BMFPN makes objects and assigned feature maps match more. And objects can also be better detected with our effective bidirectional and reticular information flows, which effectively fuses multi-level information.

B. Ablation Studies on Different Feature Pyramid Structures

To evaluate the effectiveness of our BMFPN, we compare it with different feature pyramid structures based on FCOS. It is easy to conduct by just replacing the FPN in FCOS with these different architectures. The result is shown in Table III. First, we construct a bottom-up path augmentation following PAFPN [21] and it only brings a slight improvement on AP(+0.3%). It verifies our viewpoint that each feature in the pyramid may mainly or only contain single-level information with sparse information propagation among feature layers. It is not enough

TABLE VI
COMPARISON WITH STATE-OF-THE-ART DETECTORS ON MS COCO TEST-DEV SPLIT.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage detectors							
Faster R-CNN w/FPN [17]	ResNet-101	36.2	59.1	39.0	18.2	39.0	48.2
Mask R-CNN [33]	ResNeXt-101	39.8	62.3	43.4	22.1	43.2	51.2
Cascade R-CNN [34]	ResNet-101	42.8	62.1	46.3	23.7	45.5	55.2
D-RFCN + SNIP [16]	DPN-98	45.7	67.3	51.1	29.3	48.8	57.1
TridentNet [35]	ResNet-101-DCN	46.8	67.6	51.5	28.0	51.2	60.5
One-stage detectors							
YOLOv3-608 [36]	DarkNet-53	33.0	57.9	34.4	18.3	35.4	41.9
SSD513 [7]	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8
RetinaNet800 [8]	ResNet-101	39.1	59.1	42.3	21.8	42.7	50.2
RefineDet512 [37]	ResNet-101	36.4	57.5	39.5	16.6	39.9	51.4
CornerNet511 [13]	Hourglass-104	40.6	56.4	43.2	19.1	42.8	54.3
ExtremeNet511 [14]	Hourglass-104	40.2	55.5	43.2	20.4	43.2	53.1
CenterNet511 [12]	Hourglass-104	44.9	62.4	48.1	25.6	47.4	57.4
FoveaBox [10]	ResNeXt-101	42.1	61.9	45.2	24.9	46.8	55.6
FSAF [11]	ResNeXt-101	42.9	63.8	46.3	26.6	46.2	52.7
FCOS [9]	ResNet-101	41.0	60.7	44.1	24.0	44.1	51.0
FCOS [9]	ResNeXt-101	42.1	62.1	45.2	25.6	44.9	52.0
ours	ResNet-101	43.4	62.0	46.5	24.8	46.9	55.1
ours	ResNeXt-101	44.7	63.6	48.4	26.1	48.5	57.1

to propagate low-level information to deep layers with a single information flow like bottom-up path augmentation do.

Next, we build xNets [27] following the original settings in their paper. The result shows that improvement on AP is 1.2% over baseline. Compared with xNets, our BMFPN brings another 1.8% AP improvement. It even improves AP_L by 2.1%. And it obtains more accurate detection results with improvement AP_{75} of 2.1%. Our BMFPN construct a more effective feature pyramids. The semantic and localization information of each layer are enhanced through multiple top-down and bottom-up pathways, which is what xNets lacks.

We further conduct another experiment to verify the effectiveness of our bidirectional and reticular information flow, which effectively fuses multi-level information. We directly select the diagonal matrix layers of BMFPN when making a prediction during training and inference, other layers are only used to propagate information without making further model predictions. As shown in III, the result (BMFPN*) still achieve 1.3% higher AP than baseline. Besides, our models only introduce negligible parameters as shown in the last column of Table III.

C. Ablation Studies on Connection Mode

TDM and BUM are very flexible modules in connection mode. Any one of them can be the first in series. And they can also be connected in parallel. Under parallel mode, the base features are fed into TDM and BUM respectively, and then final feature pyramid is constructed by the outputs of these two modules with element-wise addition operations. As shown in Table IV, TDM followed BUM performs best. We use this connection mode in all other experiments. Besides, we compare different connection layers between TDM and BUM in Table V. It achieves 39.7% AP with dilated convolutions. We view this as the our default setting unless specified.

D. Comparison with State-of-the-art Detectors

We evaluate our final models on the MS COCO test-dev split to compare with recent state-of-the-art methods. In order to make a fair comparison, we directly make use of all hyper-parameters of FCOS. We argue that the performance can be much improved if the hyper-parameters are tuned for our detector. Table VI presents the comparison. Our models achieve competitive performance with both one-stage and two-stage detectors. Following FCOS [9], scale jitter is used and the models are trained for $2\times$ longer iteration numbers. Without bells and whistles, our method gains +2.4% and +2.6% AP on MS COCO dataset from baselines with ResNet-101 and ResNeXt-101 backbones, respectively. With ResNet101, it even performs 1.3% AP higher than FCOS with ResNeXt-101, which is a deeper backbone. To our best knowledge, it outperforms recent state-of-the-art non-keypoint based anchor-free detection methods like FoveaBox, FSAF and FCOS.

V. CONCLUSION

In this paper, the popular feature pyramids used in advanced object detectors is reinvestigated. It is found that a poor match between rectangular objects and feature maps exists in these approaches, thus preventing from accurate recognition and location. Besides, we reveal that a sparse information flow among each feature in the pyramid can not

provide representative enough information for classification and localization sub-tasks. Bidirectional Matrix Feature Pyramid Network (BMFPN) is proposed to address these issues. It constructs a feature pyramid whose receptive fields have various scales and aspect ratios. And it forms bidirectional and reticular information flow, which effectively fuses multi-level information. To evaluate our proposed architecture, an end-to-end anchor-free detector is designed and trained by integrating BMFPN into FCOS. And the center-ness branch in FCOS is modified with our Gaussian center-ness branch (GCB), which brings another slight improvement. Extensive experiments demonstrate the effectiveness of the proposed architecture and the novel modules.

Acknowledgments: This work was partially financially supported by the National Natural Science Foundation of China under grants 61533012, 91748120 and 52041502.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representation*, 2015.

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

[4] M. Everingham and J. Winn, "The pascal visual object classes challenge 2012 (voc2012) development kit," *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, vol. 8, 2011.

[5] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," *Proceedings of the European Conference on Computer Vision*, pp. 740–755, 2014.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *Proceedings of the European Conference on Computer Vision*, pp. 21–37, 2016.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017.

[9] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9627–9636, 2019.

[10] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 840–849, 2019.

[11] T. Kong, F. Sun, H. Liu, Y. Jiang, and J. Shi, "Foveabox: Beyond anchor-based object detector," *IEEE Transactions on Image Processing*, 2020.

[12] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6569–6578, 2019.

[13] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," *Proceedings of the European Conference on Computer Vision*, pp. 734–750, 2018.

[14] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 850–859, 2019.

[15] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," *Advances in neural information processing systems*, pp. 2277–2287, 2017.

[16] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection snip," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3578–3587, 2018.

[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, 2017.

[18] P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu, "Scale-transferrable object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 528–537, 2018.

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

[20] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance problems in object detection: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[21] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, 2018.

[22] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-cnn: Towards balanced learning for object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 821–830, 2019.

[23] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018.

[24] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2det: A single-shot object detector based on multi-level feature pyramid network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9259–9266, 2019.

[25] Q. Z. S. X. Chaoxu Guo, Bin Fan and C. Pan, "Augfpn: Improving multi-scale feature learning for object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12 595–12 604, 2020.

[26] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Nas-fpn: Learning scalable feature pyramid architecture for object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7036–7045, 2019.

[27] A. Rashwan, A. Kalra, and P. Poupart, "Matrix nets: A new deep architecture for object detection," *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019.

[28] Y. Fisher and K. Vladlen, "Multi-scale context aggregation by dilated convolutions," *International Conference on Learning Representations*, 2016.

[29] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 4898–4906, 2016.

[30] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," *Proceedings of the 24th ACM International Conference on Multimedia*, pp. 516–520, 2016.

[31] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[32] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 658–666.

[33] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, 2017.

[34] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, 2018.

[35] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-aware trident networks for object detection," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6054–6063, 2019.

[36] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[37] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4203–4212, 2018.