Selective Feature Network for Object Detection

Yuning Cui[1], Dianxi Shi[2,3,*] , Yongjun Zhang[2], Qianchong Sun[2]
[1]*College of Computer, National University of Defense Technology, Changsha, China*
[2]*Artificial Intelligence Research Center (AIRC), National Innovation Institute of Defense Technology (NIIDT), Beijing, China*
[3]*Tianjin Artificial Intelligence Innovation Center (TAIIC), Tianjin, China*
{cuiyuning16, dxshi, yjzhang}@nudt.edu.cn, sunqianchong@hotmail.com

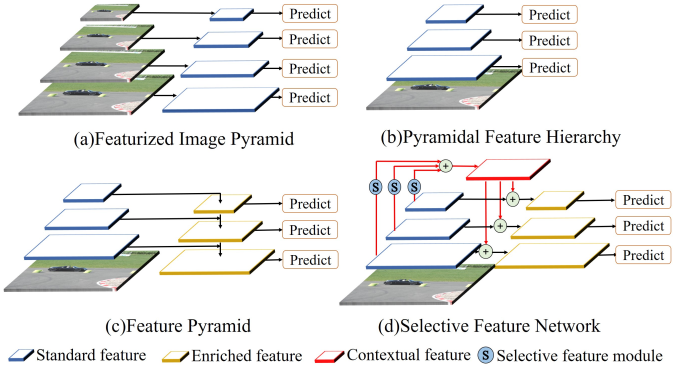
***Abstract*—Scale variation is one of the important challenges in object detection. Many state-of-the-art objectors tackle this problem by utilizing the feature pyramids. However, the current methods of producing feature pyramids are still inefficient to integrate the semantic information from other layers. In this work, our motivation is to build a feature pyramid efficiently with the selected contextual feature by integrating the informative features and suppressing the useless ones. To achieve this goal, we propose a novel single-stage detection network termed Selective Feature Network(SFNet) which consists of a semantic-enhanced module and a selective feature module. The semantic-enhanced module improves the semantics of basic pyramids via a light-weight architecture. In conjunction with that, a selective feature module is employed to combine features across different channels and scales by attention mechanism. The resulting contextual feature is then injected into the pyramidal features. Comprehensive experiments are performed on PASCAL VOC and MS COCO datasets. Results demonstrate that, with a VGG16 based SFNet, our approach obtains significant improvements over the competitors without losing real-time processing speed.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

In recent years, with the emergence of convolutional neural networks(CNNs), the great success has been achieved in the compute vision tasks such as image classification [1], [2], object detection [3], [4] and semantic segmentation [5], [6]. With a wide range of applications, the object detection as a fundamental task has been extensively studied. Currently, these CNN-based object detection frameworks can be divided into two categories: the one-stage methods such as SSD [7] or YOLO [4] and the two-stage methods such as Faster-RCNN [8] or R-FCN [9].The two-stage methods extract proposals first and then perform classification and regression on them. The one-stage methods directly predict the bounding boxes by dense grids on the input image. Generally, two-stage methods have the advantage of being more accurate while one-stage methods obtain a real-time processing speed but compromise on performance. However, scale variation is one of the key challenges for both methods.

To solve this issue, a traditional way is to build multi-scale image pyramids as shown in Figure 1(a). This intuitive way has been applied for both CNN-based methods [10] and methods along with hand-crafted features. Nevertheless, this kind of approaches are quite inefficient and infeasible for practical applications due to the increase of inference time.



(a)Featurized Image Pyramid

(b)Pyramidal Feature Hierarchy

(c)Feature Pyramid

(d)Selective Feature Network

Fig. 1. Different strategies for constructing feature pyramid. (a) Using an images with various sizes to build a feature pyramid. (b) Pyramidal feature hierarchy computed by a ConvNet as if it were a featurized image pyramid. (c)Feature pyramid network utilizes the features generated by top-down pathway and lateral connections for prediction. (d) Our proposed Selective Feature Network(SFNet) builds a feature pyramid effectively by selecting informative features via attention mechanism.

We focus on the CNN-based methods which can approximate the image pyramids with less computation consumption. SSD is one of the first attempts exploring the feature pyramids in the deep learning era. SSD takes a truncated VGG16 as base network and adds a series of convolutional networks to generate further feature maps. Based on that, several object detection feature maps with varying sizes are built. Given a single input image, SSD utilizes multi-scale feature maps to conduct independent predictions (Figure 1(b)). Shallow layers with high-resolution feature maps are responsible for small objects while deep layers with high-level semantic information are for large objects. However, the original SSD struggles to tackle the scale variation problem since the former layers fail to capture the rich semantics. This impedes SSD from detecting small instances so SSD still lags behind the-state-of-art detectors in terms of accuracy.

To tackle the problem mentioned above in SSD, many recent works [11], [12] integrate semantic information at all scales to improve the performance of small object detection and make the framework more robust to object scales. In Figure 1(c), Feature pyramid network(FPN) [13] and RetinaNet [41] enrich the former layers with features through lateral connections in top-down pathway. The lateral connections pass the high-level semantic information from deep features to shadow features

*Corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

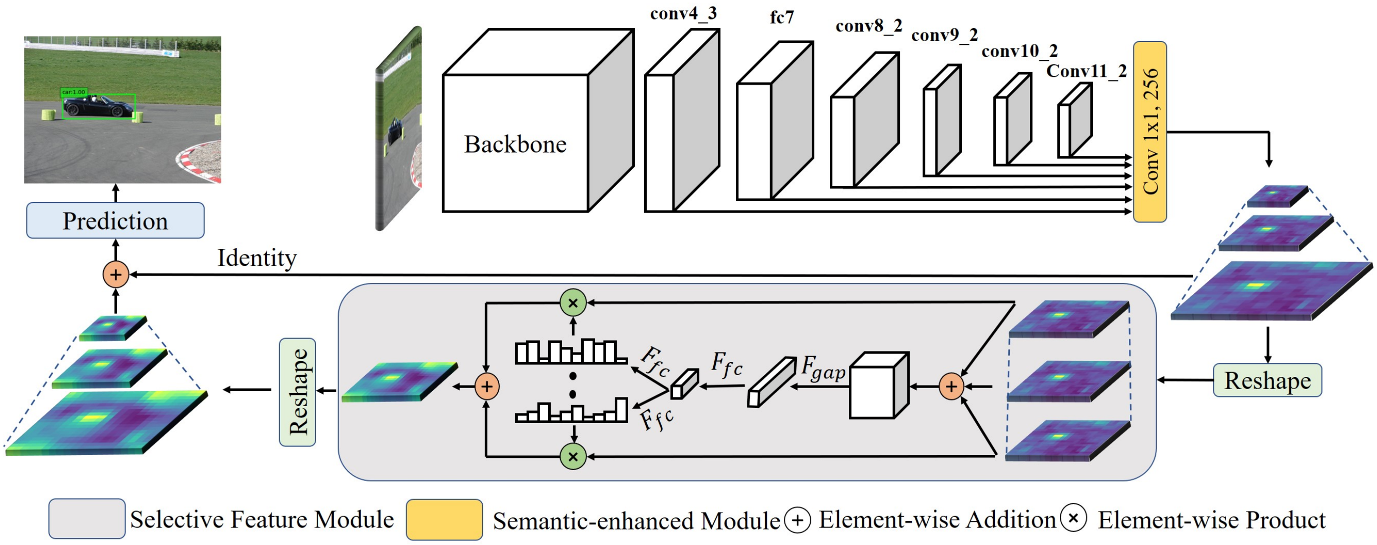


Fig. 2. Overall architecture of our one-stage detector based on VGG. It mainly consists of three components: backbone network, semantic-enhanced module and selective feature module. Semantic-enhanced module is designed to improve the semantics of low level pyramid. Selective feature module is added to inject the informative features into pyramid across different spatial locations and scales via attention mechanism, shown in the low half.

layer by layer until the finest resolution map is generated.

FPN reuses multi-scale features maps within a single network and enriches the semantic information via lateral connections to boost the performance of object detection. However, the shallow layers in FPN only focus more on adjacent resolution but less on others. As a result, the deeper layers fail to integrate with the shallower layers. Our experimens shows that shallow layers are also helpful to improve the accuracy. Further, shallow layers take in other features entirely without selection, which leads to the important features covered up by the useless ones.

In this paper, we aim at solving the obstacles discussed above by designing two modules, a semantic-enhanced module and a selective feature module. Our motivation is to strengthen the prediction layers with the most useful semantic information generated by attention mechanism. To be specific, the semantic-enhanced module is employed to improve the nonlinearity of the low level features and change the number of channels. Further, to enable the network select more significant semantic information in a flexible way, we introduce a selective module as is shown in Figure 1(d). For varying size of input image, this module utilizes attention mechanism to emphasize the favorable information from different scales and spatial locations. Then, the resulting contextual feature is formed and instilled into the original pyramid layers for prediction. As a result, every layer obtains strong semantics from all layers not only the adjacent one. Compared to the previous works such as standard SSD and FPN frameworks, our approach have three advantages over them: (1) our method introduces the global semantics for low-level pyramids; (2) meanwhile, this effective method prunes out the useless information and generate the appropriate semantics for scale variation of input images; (3) the contextual semantics are integrated into all scales simultaneously, which is efficient than the lateral connections.

We do comprehensive experiments upon the diverse feature pyramid strategies using VGG16 base network and the results demonstrate that our approach is more competitive in aspects of accuracy and speed.

We summarize our contributions as follows:

- 1) We introduce a effective feature pyramid strategy consisting of semantic-enhanced module and selective feature module to enrich the pyramids with the appropriate contextual semantics.
- 2) We compare some popular feature pyramid strategies with our method within the standard SSD architecture in terms of accuracy, efficiency or parameter size, and the experimental results show that our method has an advantage over them.
- 3) Our time efficient method improves the performance compared with popular single-stage detectors by a large margin on both PASCAL VOC and MS COCO datasets.

II. Related Work

Deep object detectors. General object detection is a fundamental task in the domain of computer vision and has been extensively studied. Recently, the object detectors based on deep learning have obtained dramatical improvement in both accuracy and speed. In terms of architecture, CNN-based object detectors can be divided into two types. The first type is two-stage detectors. These detectors generates a pool of object proposals and refine them. Particularly, RCNN [15] employs Selective Search [16] to generate region proposals and then do classification and regression on the cropped proposals independently. To avoid extracting feature repeatedly for a single image, SPP-Net [17] and Fast R-CNN do feature extraction only once upon the whole input image. These two methods employ spatial pyramid pooling or ROI pooling to generate region features and allow the reuse of the feature maps. Further, Faster R-CNN [8] proposes Region Proposal Network(RPN) sharing the network with the detection backbone network to replace the time-consuming region proposal step. And Faster R-CNN is a complete end-to-end detection framework. R-FCN proposes a fully convolutional network for object detection. The second type is one-stage detectors. To speed up the inference, YOLO and SSD are proposed. These detectors eliminate the proposal generation step and directly do classification and bounding box regression on the pre-defined anchors. In this paper, we base our approach on standard SSD due to its speed/accuracy tradeoff compared with two-stage detectors.

Feature pyramids for tackling scale variation. As one of the key challenges in computer vision tasks, scale variation has received lots of attention. To tackle this problem, SSD generates default boxes at different depth layers and perform object detection at multiple features without fusion. This conduction leads to limited semantic information in shallow layers to detect small instances. Many published literature [18], [19]and [20] solve this issue by extracting better features or exploiting contextual information. A very popular strategy is to build top-down feature pyramid presentation and utilize lateral connections to convey the high-level semantics to former layers. After that, PANet [21] builds a bottom-up pathway to inject the low-level information into deep layers. Recently, a reconfiguration architecture is proposed to combine low-level representations with high-level semantic features in a highly-nonlinear way [22]. And the method in [23] strengthens the multi-level features using the balanced semantic features instead of lateral connections.

Attention for visual tasks. Attention has been incorporated in feed-forward convolutional neural networks. Generally speaking, visual attention mechanism enforces model to adaptively focus on the more important features. In the image classification domain, the Squeeze-and-Excitation block in SENet [24] can be considered as the channel attention and it is developed to model interdependencies between channels. The SKNet [25] can be regarded the kernel attention to yield different sizes of receptive fields. There are many meaningful works introducing attention into object detection networks. DFPR [22] applies the Squeeze-and-Excitation block as the basic module to emphasize useful features and suppress less useful ones. CBAM [26] sequentially applies channel attention and spatial attention for emphasizing meaningful features. Further, HAR-Net [27] proposes hybrid attention modules consisting of channel attention, spatial attention and aligned attention for single-stage object detection.

Most of above-mentioned methods apply attention mechanism in the backbone network. To the best of our knowledge, it is far from development to build more effective pyramid features via attention mechanism. Our proposed method utilizes attention mechanism to adaptively select useful features from multiple layers to build feature pyramid.

III. Method

Here, we will introduce the overall architecture of the proposed method and describe the semantic-enhanced module and the selective feature module. The overall of our approach, named SFNet, is illustrated in Figure 2. SFNet mainly consists of there parts: a base network for feature extraction, a semantic-enhanced(SE) module and a selective feature(SF) module to produce feature pyramid representation.

We employ VGG16 architecture as in [7]. To deal with the problems discussed previously, the semantic-enhanced module add non-linearities to improve the representation power first. Then the SF module prunes out the useless information of features produced by SE module and enables model to focus on the most useful semantic information for object detection. Finally, the selected feature is fused with the previous pyramidal features. The resulting feature pyramid captures rich semantic information for detection.

A. Backbone Network

We build our method on the standard SSD system due to its better accuracy-vs-speed tradeoff. The standard SSD detector employs a VGG16 as the backbone network. From original VGG16 architecture, SSD uses the conv4_3 layer and converts the fc_7 to convolutional layer for detection. For a given input size of 300×300, the above layers generate 38×38 feature map and 19×19 feature map which have low-level semantic information for object detection. Based on that, to detect instances from small to large, SSD truncates the last fully connected layer of original VGG16 and builds a series of conv layers. The additional prediction layers are conv8_2, conv9_2, conv10_2 and conv11_2 with feature map size of 10×10, 5×5, 3×3, 1×1, respectively. Meanwhile, SSD spreads out anchors with different scales and aspect ratios to multiple layers of different depths. It forms the pyramidal hierarchical structure where former layers are responsible for small objects while later layers for large objects.

Following the original SSD[7], we utilize six prediction layers for 300×300 input images size and seven for 512×512 input images size, instead of reduced layers in [39]. Decreasing the prediction layers will lead to the performance degraded in our model.

B. Micro Semantic-enhanced Module

To select appropriate feature for building pyramidal feature structure, the precondition is that the pool for selecting has enough information. Nevertheless, the basic feature maps generated by backbone network have low level semantic information. To improve the semantics to be learned, we utilize the lightweight 1×1 conv layer as in [33]. It takes the low level detection feature maps produced by backbone network and outputs the enhanced feature maps for each original layer respectively, as showed with yellow-box in Figure 2. Meanwhile, the SE module also changes the channel dimensions of the input layers. We set k=256 in this paper, so each pyramidal layer have 256 channels after undergoing the SE module.

Mathematically, given an input image, the outputs of backbone network are expressed as:

$$X_b = x_1, x_2, ..., x_n \tag{1}$$

where n denotes the number of features in pyramid. Then the enhanced features X_e obtained after undergoing the SE module, is formulated as:

$$X_e = f^{1\times1}(X_b) \tag{2}$$

where $f^{1\times1}$ denotes the convolutional layer with kernel size of 1x1 followed by Batch Normalization [28] and ReLU. We have experimented with more sophisticated blocks (e.g., using multi- layer residual blocks [33]) but not observed better results.

It's worth noting that the SE module is the solo additional layer to inject non-linearities into the pyramids. Simple but effective, the SE module improve the semantics to be learned of the basic pyramids.

IV. Selective Feature Module

Given the feature hierarchy, the goal of selective feature module is to select the most useful features for the pyramids by assigning a big weight to the informative feature and a small weight to useless one. In this paper, the SF module is inspired by Selective Kernel Convolution [25] which consists of three parts: Split, Fuse and Select. Here, we have obtained the feature maps with different sizes so our SF module only consists of the later two parts: Fuse and Select, as highlighted with grey-box in Figure 2 where a two-feature case is shown.

In the previous section, we have obtained pyramidal features with the same channel-dimension but different sizes. Before fusion, we first reshape the pyramids to the same spatial size. The pyramid with uniform resolution is expressed as:

$$U = u_1, u_2, ..., u_n \tag{3}$$

where, $u_i \in \mathbb{R}^{H\times W\times C}$ denotes the spatial dimension and C denotes the channel dimension.

The motivation of this paper is to select the informative features via attention mechanism. To achieve this aim, the generation of weights need to synthesize information from all input features. So, the fused feature is obtained by integrating all input features (two in Figure 2) via element-wise addition:

$$F = \sum_i u_i \tag{4}$$

Then, the global information is generated by a global average pooling(GAP) operation on each channel of F:

$$S_c = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} F_c(i,j) \tag{5}$$

where S_c is the cth value of $S \in \mathbb{R}^C$ and $F_c(i,j)$ denotes the value of each element at cth channel.

Further, a fully connected layer is followed to reduce the dimension for better efficiency and precise selection. The low-dimension feature is created:

$$z = \delta(\sigma(W_1 S) \tag{6}$$

where δ is the ReLU function, σ is Batch Normalization and $W_1 \in \mathbb{R}^{d\times C}$ .In this paper, d is set to 32 to make dimensionality-reduction.

In the select stage, to select informative features for detection, a series of fully connected layers are employed to produce the primary weight vector for each original feature:

$$a_i = W_{2i} z \quad i = 1, 2, ..., n \tag{7}$$

where a_i refers to the i th weight vector, $W_{2i} \in \mathbb{R}^{d\times C}$.

Then, softmax operation is performed on the same index of every primary weight vector to adaptively select features from different branches:

$$Q_{ij} = \frac{e^{a_{ij}}}{\sum_i e^{a_{ij}}} \quad j = 1, 2, ..., k \tag{8}$$

where, Q_i is the final weight vector for the ith feature in pyramid, j specifies jth element of the ith weight.

To obtain the weighted feature, we perform channel-wise multiplication on final weights and features of uniform size. After that, the contextual feature G is produced:

$$G = \sum_i Q_i \otimes U_i \tag{9}$$

where $\otimes$ refers to channel-wise multiplication.

The contextual feature integrates the informative features and suppress the useless ones via attention mechanism. To instill contextual feature into low level features of different resolution, G is reshaped into corresponding size with different depth of pyramid. The reshaped features form the contextual pyramid expressed as $\{G_1, G_2, \ldots, G_n]\}$. In the end, the prediction layers L are produced through element-wise summation on contextual pyramid and low level pyramid:

$$l_i = G_i + u_i \tag{10}$$

It should be noted that the original selective kernel convolution is developed to enable the neurons to adaptively adjust their receptive fields sizes for image classification task. In contrast, we apply it to select informative features and suppress useless feature for object detection. Given a low level feature pyramid, the selective kernel convolution helps to pick out the meaningful features. Our method is also different from the approach in [22] which applies the Squeeze-and-Excitation block [24] as the basic module. Result (click ”Generate” to refresh) Copy to clipboard

V. Experiments

We conduct comprehensive experiments on two widely used datasets: PASCAL VOC [29] and MS COCO [30]. In this section, we first introduce the datasets and give an implementation details of our method. Then we make comparison with existing object detectors and provide ablation study on the PASCAL VOC2007 dataset.

TABLE I

Per-class comparison on the PASCAL VOC 2007 test set. All models are trained on VOC2007 trainval and VOC2012 trainval. The first section contains some representative two-stage detectors with large input size. The second section and the last section contain the results of single-stage detector with different sizes input images. SFNet300 indicates the input image dimension is 300 × 300.

Method	**Backbone**	**mAP**	**aero**	**bike**	**bird**	**boat**	**bottle**	**bus**	**car**	**cat**	**chair**	**cow**	**table**	**dog**	**horse**	**mbike**	**person**	**plant**	**sheep**	**sofa**	**train**	**tv**
Faster [8]	VGG16	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
Faster [33]	ResNet101	76.4	79.8	80.7	76.2	68.3	55.9	85.1	85.3	89.8	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	85.3	72.0
R-FCN [9]	ResNet101	80.5	79.9	87.2	81.5	72.0	69.8	86.8	88.5	89.8	67.0	88.1	74.5	89.8	90.6	79.9	81.2	53.7	81.8	81.5	85.9	79.9
SSD300 [7]	VGG16	77.5	79.5	83.9	76.0	69.6	50.5	87.0	85.7	88.1	60.3	81.5	77.0	86.1	87.5	83.9	79.4	52.3	77.9	79.5	87.6	76.8
STDN300 [34]	DenseNet	78.1	81.1	86.9	76.4	69.2	52.4	87.7	84.2	88.3	60.2	81.3	77.6	86.6	88.9	87.8	76.8	51.8	78.4	81.3	87.5	77.8
DSSD321 [11]	ResNet101	78.6	81.9	84.9	80.5	68.4	53.9	85.6	86.2	88.9	61.1	83.5	78.7	86.7	88.7	86.7	79.7	51.7	78.0	80.9	87.2	79.4
DFPR300 [22]	VGG16	79.6	84.5	85.5	77.2	72.1	53.9	87.6	87.9	89.4	63.8	86.1	76.1	87.3	88.8	86.7	80.0	54.6	80.5	81.2	88.9	80.2
DES300 [35]	VGG16	79.7	83.5	86.0	78.1	74.8	53.4	87.9	87.3	88.6	64.0	83.8	77.2	85.9	88.6	87.5	80.8	57.3	80.2	80.4	88.5	79.5
SFNet300(ours)	VGG16	**79.9**	84.5	87.2	78.7	73.9	56.2	88.2	87.2	87.7	63.4	85.9	77.4	86.5	88.6	87.8	80.7	57.3	79.6	80.2	88.1	78.3
SSD512 [7]	VGG16	79.5	84.8	85.1	81.5	73.0	57.8	87.8	88.3	87.4	63.5	85.4	73.2	86.2	86.7	83.9	82.5	55.6	81.7	79.0	76.6	80.0
SSD513 [7]	ResNet101	80.6	84.3	87.6	82.6	81.6	59.0	88.2	88.1	89.3	64.4	85.6	76.2	88.5	88.9	87.5	83.0	53.6	83.9	82.2	87.2	81.3
DSSD513 [11]	ResNet101	81.5	86.6	86.2	82.6	74.9	62.5	89	88.7	88.8	65.2	87	78.7	88.2	89	87.5	83.7	51.1	86.3	81.6	85.7	83.7
STDN513 [34]	DesnseNet	80.9	86.1	89.3	79.5	74.3	61.9	88.5	88.3	89.4	67.4	86.5	79.5	86.4	89.2	88.5	79.3	53.0	77.9	81.4	86.6	85.5
DFPR512 [22]	VGG16	81.1	90.0	87.0	79.9	75.1	60.3	88.8	89.6	89.6	65.8	88.4	79.4	87.5	90.1	85.6	81.9	54.8	79.0	80.8	87.2	79.9
SFNet512(ours)	VGG16	**81.6**	87.8	87.6	84.7	74.7	65.3	88.4	88.9	88.4	65.7	88.4	75.8	86.3	89.1	86.4	84.0	57.9	84.6	78.4	86.9	81.7

TABLE II

PASCAL 2012 detection results. Note that all models in this table are trained on VOC2007 trainvaltest and VOC2012 trainval.

Method	**Backbone**	**mAP**	**aero**	**bike**	**bird**	**boat**	**bottle**	**bus**	**car**	**cat**	**chair**	**cow**	**table**	**dog**	**horse**	**mbike**	**person**	**plant**	**sheep**	**sofa**	**train**	**tv**
Faster [33]	ResNet101	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
R-FCN [9]	ResNet101	77.6	86.9	83.4	81.5	63.8	62.4	81.6	81.1	93.1	58.0	83.8	60.8	92.7	86.0	84.6	84.4	59.0	80.8	68.6	86.1	72.9
ION [36]	VGG16	76.4	87.5	84.7	76.8	63.8	58.3	82.6	79.0	90.9	57.8	82.0	64.7	88.9	86.5	84.7	82.3	51.4	78.2	69.2	85.2	73.5
SSD300 [7]	VGG16	75.8	88.1	82.9	74.4	61.9	47.6	82.7	78.8	91.5	58.1	80.0	64.1	89.4	85.7	85.5	82.6	50.2	79.8	73.6	86.6	72.1
SSD321 [7]	ResNet101	75.4	87.9	82.9	73.7	61.5	45.3	81.4	75.6	92.6	57.4	78.3	65.0	90.8	86.8	85.8	81.5	50.3	78.1	75.3	85.2	72.5
DES300 [35]	VGG16	77.1	88.5	84.4	76.0	65.0	50.1	83.1	79.7	92.1	61.3	81.4	65.8	89.6	85.9	86.2	83.2	51.2	81.4	76.0	88.4	73.3
DSSD321 [11]	ResNet101	76.3	87.3	83.3	75.4	64.6	46.8	82.7	76.5	92.9	59.5	78.3	64.3	91.5	86.6	86.6	82.1	53.3	79.6	75.7	85.2	73.9
DFPR300 [22]	VGG16	77.5	89.5	85.0	77.7	64.3	54.6	81.6	80.0	91.6	60.0	82.5	64.7	89.9	85.4	86.1	84.1	53.2	81.0	74.2	87.9	75.9
SFNet300(ours)	VGG16	**77.6**	89.9	85.3	76.4	64.1	52.7	83.9	79.3	91.7	61.2	83.7	66.5	90.5	87.7	86.4	83.9	53.9	82.3	73.7	86.5	73.3
SSD512 [7]	VGG16	78.5	90.0	85.3	77.7	64.3	58.5	85.1	84.3	92.6	61.3	83.4	65.1	89.9	88.5	88.2	85.5	54.4	82.4	70.7	87.1	75.6
SSD512 [7]	ResNet101	79.4	90.7	87.9	78.3	66.3	56.5	84.1	83.7	94.2	62.9	84.5	66.3	92.9	88.6	87.9	85.7	55.1	83.6	74.3	88.2	76.8
DSSD513 [11]	ResNet101	80.0	92.1	86.6	80.3	68.7	58.2	84.3	85.0	94.6	63.3	85.9	65.6	93.0	88.5	87.8	86.4	57.4	85.2	73.4	87.8	76.8
DFPR512 [22]	VGG16	80.0	89.6	87.4	80.9	68.3	61.0	83.5	83.9	92.4	63.8	85.9	63.9	89.9	86.2	56.3	84.4	75.5	89.7	78.5	89.7	78.5
RefineDet [37]	VGG16	80.1	90.2	86.8	81.8	68.0	65.6	84.9	85.0	92.2	62.0	84.4	64.9	90.6	88.3	87.2	87.8	58.0	86.3	72.5	88.7	76.6
SFNet512(ours)	VGG16	**80.3**	90.9	87.7	80.6	67.5	61.9	85.6	85.3	92.2	64.4	86.2	65.3	90.7	90.5	88.7	87.7	57.9	85.4	72.2	88.4	76.9

A. *Dataset*

The PASCAL VOC dataset contains 20 different object classes. For VOC 2007, training is performed on the union of VOC 2007 trainval and VOC 2012 trainval and we use VOC 2007 test set for evaluating. For VOC 2012, models are trained on the union of VOC 2007 trainval, 2007 test and 2012 trainval and evaluated on VOC2012 test. For evaluation, the standard mean average precision(mAP) is used.

The COCO dataset is more challenging consisting of natural images from 80 object categories. For COCO, we use a popular split where training is performed on trainval35k, validating on minival with 5k images and we evaluate our method on the official test-dev 2017 evaluation server.

B. *Implementation Details*

We implement our SFNet detector based on Pytorch framework [27]. We follow most settings as SSD, including scales and aspect ratios of the defaults boxes, data augmentation and loss functions. The warm-up strategy is adopted that the learning rate linearly increases from 10-6 to 4×10-3 at the first 5 epochs. Then, the learning rate is divided by 10, for PASCAL VOC dataset at 150 and 200 epoch, and for MS COCO dataset at 90 and 120 epoch. The total numbers of training epochs are 250 and 140 for VOC dataset and COCO dataset, respectively. We set weight decay to 0.0005, momentum to 0.9 and batch size to 32 for all of experiments. We initialize all new layers with the MSRA method [32]. For VGG16 backbone, we choose the spatial size of conv8_2 as the uniform size in SF module.

C. *PASCAL VOC*

We compare our method with the baseline SSD and other existing detectors. Table I shows the per-class results for varying input image sizes on the PASCAL VOC 2007 test set. SSD uses features with different depths performing detection independently and achieves a score of 77.5 for low resolution and 79.5 for high resolution. For the images resolution, SFNet300 scores 79.9%, 2.4% higher than that of SSD300 and 2.1% higher than that of SSD512. As the upgraded version of SSD, DSSD [11] replaces the backbone network with deep residual network [33], which will yield features

TABLE III
DETECTION RESULTS ON COCO 2017 TEST-DEV

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster [8]	VGG16	21.9	42.7	-	-	-	-
R-FCN [9]	ResNet101	29.2	51.5	-	10.3	32.4	43.3
SSD300 [7]	VGG16	25.1	43.1	25.8	6.6	25.9	41.4
RON384++ [12]	VGG16	27.4	49.5	27.1	-	-	-
SSD321 [7]	ResNet101	28.0	45.4	29.3	6.2	28.3	49.3
FSSD300 [18]	VGG16	27.1	47.7	27.8	8.7	29.2	42.2
DSSD321 [11]	ResNet101	28.0	46.1	29.2	7.4	28.1	47.6
STDN300 [34]	DenseNet	28.0	45.6	29.4	7.9	29.7	45.1
SFNet300(ours)	VGG16	**28.1**	47.6	29.1	10.3	29.9	42.7
SSD512 [7]	VGG16	28.8	48.5	30.3	10.9	31.8	43.5
SFNet512(ours)	VGG16	**31.7**	52.5	33.4	15.1	34.6	44.8

with high-level semantics. Our method performs better than DSSD for various sizes of input images. SFNet300 provides a gain of 1.3% in terms of mAP, over DSSD321. It is worth mentioning that DFPR [22] shares the same motivation with our method but SFNet is 0.5% better than it with an input image size of 512×512. Furthermore, SFNet still achieves a significant improvement compared with other popular single-stage detectors.

All the results demonstrate that SFNet has integrated more useful information. As shown in Table II, the same conclusion can be drawn for VOC 2012 test set.

D. MS COCO

We also evaluate the performance of our method on the MS COCO dataset. Table III shows the comparison results on COCO test-dev 2017 from the official evaluation server. Compared to baseline SSD, our detector obtains better performance on all of the metrics.

With the same input image size, SFNet300 achieves a significant improvement of 12% on the over-all detection performance, from 25.1 to 28.1. For small objects, our method outperforms the baseline SSD300 with a large margin. This again demonstrates that the prediction layers especially the shallow layers capture the useful information from SF module. It is noteworthy to mention that, with input image size of 321×321, SSD321 and its variation DSSD321 scores 28.0% mAP based on ResNet101 backbone and our SFN300 outperforms them with a shallower backbone VGG and lower image resolution. For the image size of 512×512, our method improves the initial SSD from 28.8 to 31.7.

VI. DISCUSSION

A. Ablation Study on PASCAL VOC 2007

To better understand the effectiveness of our detector, we conduct an ablation study on the VOC2007 test dataset based on our SFNet300. All of models in this section are trained on VOC2007 trainval and VOC2012 trainval.

The impact of the proposed modules. As can be seen in Table IV,the baseline SSD300 scores 77.5% mAP and the semantic-enhanced module(SE) improves the result by 1.3%. With the selective feature(SF) module added, the result can be further improved to 79.7%, which is 2.2% better than baseline. The results confirm our intention that building feature pyramid by selective feature contributes to a better detector.Then, we switch the integration strategy from summation to element-wise product in (9), but it can't achieve better performance, illustrated in the fifth row of Table IV. Further, we investigate another two popular strategies to construct the feature pyramid. The forth column of Table V shows the comparison in terms of accuracy when using different feature pyramid strategies. The strategy as in [23], which strengthens the multi-level features via summation of original pyramids, obtains a detection mAP score of 79.3 without refinement operation, illustrated in the forth row of Table V. The table also shows the result of SSD with lateral connections. As is shown in the table, our strategy surpasses the alternatives.

TABLE IV
ABLATION RESULTS ON VOC 2007 TEST DATASET. SE IS SEMANTIC-ENHANCED MODULE. SF IS SELECTIVE FEATURE MODULE.

Method	mAP
Baseline SSD300	77.5
SSD300+SE	78.8
SSD+SE+SF	79.7
SSD+SE+SF(mul)	79.6
SSD+SE+SF(add/only scales)	79.3
SSD+SE+SF(add)	79.9

TABLE V
ALTERNATIVES FOR FEATURE PYRAMID STRATEGIES. FOR SSD, WE RE-IMPLEMENT THE MODEL IN PYTORCH 0.4.0.

Method	Backbone	FPS	mAP
SSD(ours-re)	VGG16	106.4	77.5
SSD+lateral	VGG16	94	78.3
SSD+sum	VGG16	99	79.3
SFNet300	VGG16	98	79.9
SFNet512	VGG16	50.3	81.6

To validate that SF module gathers features across two dimensions: spatial locations and scales, we design a experiment where each channel in a feature with specific scale shares the same weight. It means that there is only one weight to be learned for each feature in pyramid. The second line from the bottom of Table IV shows that the model only gets 79.3% mAP without channel-wise weight.

Further, we investigate another two popular strategies to construct the feature pyramid. TABLE V shows the comparison in terms of accuracy when using different feature pyramid strategies. The strategy as in [23], which strengthens the multi-level features via summation of original pyramids, obtains a detection mAP score of 79.3 without refinement operation. The table also shows the result of SSD with lateral connections. As is shown in the table, our strategy surpasses the alternatives.

Number of feature maps for selecting. To validate how the number of feature maps influences the model performance with SF module plugged in, we design a group of experiments by setting different numbers of feature maps for selecting. To reduce the number of combinations, the middle two layers for prediction, consisting of conv8_2 and conv9_2, are involved

TABLE VI
ALTERNATIVES FOR FEATURE PYRAMID STRATEGIES. FOR SSD, WE RE-IMPLEMENT THE MODEL IN PYTORCH 0.4.0.

	Feature				Size			mAP
	con6_2	fc7	con10_2	conv11_2	5	10	19	
SSD								77.5
(a)	✓	✓	✓			✓		79.4
(b)	✓	✓		✓		✓		79.6
(c)	✓	✓				✓		79.2
(d)		✓	✓	✓		✓		79.8
(e)	✓		✓	✓		✓		79.5
(f)			✓	✓		✓		79.5
(g)	✓	✓	✓	✓	✓			79.5
(h)	✓	✓	✓	✓		✓		**79.9**
(i)	✓	✓	✓	✓			✓	**79.9**

TABLE VII
RESULTS USING MOBILENET AS THE BACKBONE

Method	Backbone	mAP	Size(M)
SSD(ours-re)	MobileNet v1	69.7	5.69
ours	MobileNet v1	71.3	6.39

in all experiments and they are not shown in the TABLE VI. The experiments show that every layers in the table is essential for boosting the results especially the deeper layers. With the absence of conv11_2 layer, the accuracy drops by a large margin(-0.5%). With more deeper feature maps absent, the score further decreases. As a summary, each layer in feature pyramid contributes to obtaining a better selective feature in SF module.

Different sizes for integration. Different sizes for integration. In the pipeline of SF module, each layer is reshaped to a uniform size for integration. Here, we design a group of experiments to integrate feature pyramid into different sizes. The last three lines in TABLE VI show the results. We find that it has no influence with a large version of integration size while using a small size drops accuracy from 79.9% to 79.5%. We think the difference is that when most of shallow layers reshaped to a small size, the useful information is lost. With the same result for sizes of 19×19 and 10×10, we choose smaller one in consideration of efficiency.

Applied to lightweight backbone. We use reduced VGG16 as our backbone, but it still has more parameters compared with the lightweight network, e.g., MobileNet [14] and ShuffleNet [38]. To further validate the generalization ability of our method, we apply our method to MobileNet-SSD. Here, we choose the spatial size of conv_13 as the uniform size in SF module. TABLE VII shows that our method improves the performance significantly with the limited additional parameters. This implies that our method has potential applications for mobile devices.

B. Inference Speed

To quantitatively test the speed, we run all models in Table V using an Nvidia 1080Ti, cuda 8.0 and cuDNN v7 with Inter Xeon Silver 4110@2.10GHz.

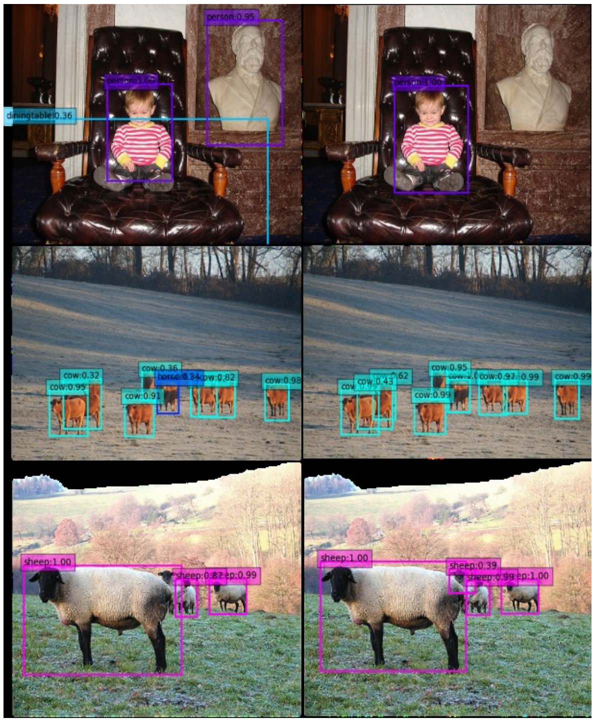


Fig. 3. Detection examples on VOC 2007 test. Left: SSD300. Right: SFNet300.

To make fair comparison, the speed is evaluated with the same batch size. We reimplement SSD using Pytorch and the accuracy is the same as reported in [7]. All results are shown in the third column of TABLE V. SFNet300 has an FPS of 98 with an accuracy of 79.9. Compared with lateral connections, our method obtains higher accuracy and faster speed. This is mainly due to the fact that the feature pyramid is generated serially with lateral connections . It is more efficient to built pyramids simultaneously in our model. At high resolution, SFNet512 achieves 81.6% mAP without losing real-time speed.

C. Detection Examples

Figure 3 shows some detection examples on VOC 2007 test set for SSD300 and SFNet300. We show ‘person’ in the first row and animals(cows&sheep) in the last two rows. Each color is associated with an object category in that image. From the results, we can see that our method can prune out the false positives which are incorrectly viewed as dining table and person categories in the first row. Compared with baseline, our model is also better at detecting small objects and occluded objects like small cows and occluded sheep.

VII. CONCLUTION

In this paper, we propose a novel single-stage method named Selective Feature Network(SFNet). To build a feature pyramid effectively for addressing the multiscale problems, we introduce a lightweight semantic-enhanced module to improve the semantics and a selective feature module to focus on the useful features and suppress the useless ones via attention

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

mechanism. A selected contextual feature, obtained by integrating features across different scales and spatial locations, is then injected into low level detection layers. Comprehensive experiments demonstrate that our method has an advantage over competitors. We believe that our approach can also be applied to two-stage detectors or deeper backbones and we take this as our future work.

Acknowledgment

This work was jointly supported by the National Key R&D Program of China (2017YFB1001901).

References

[1] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp.4700-4708.

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

[3] P. Tang, X. Wang, X. Bai, and W. Liu,"Multiple instance detection network with online instance classifier refinement," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition*,2017,pp. 2843-2851.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-Time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*,2016, pp. 779-788.

[5] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE transactions on pattern analysis and machine intelligence*, 2018, pp.834-848.

[6] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961-2969.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y., Fu, A.C. Berg, "SSD: Single shot multibox detector," in *European conference on computer vision*, 2016, pp.21-37.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, pp.1137-1149.

[9] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, 2016, pp.379-387.

[10] E. H. Adelson, P. J. Burt, C. H. Anderson, J. M. Ogden, and J. R. Bergen, "Pyramid methods in image processing," RCA Eng., 1984, pp.33-41.

[11] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[12] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: Reverse connection with objectness prior networks for object detection," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp.5936-5944.

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

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, and T. Weyand, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp.580-587.

[16] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, 2013, pp. 154-171.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, pp.1904-1916.

[18] Z. Li and F. Zhou, "FSSD: feature fusion single shot multibox detector,"*arXiv preprint arXiv:1712.00960*, 2017.

[19] S.-W. Kim, H.-K. Kook, J.-Y. Sun, M.-C. Kang, and S.-J. Ko, "Parallel feature pyramid network for object detection," in *Proceedings of the European Conference on Computer Vision*,2018, pp. 234-250.

[20] H. Li, Y. Liu, W. Ouyang, and X. Wang, "Zoom out-and-in network with map attention decision for region proposal and object detection," *International Journal of Computer Vision*, 2019, pp.225-238.

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

[22] T. Kong, F. Sun, W. Huang, and H. Liu, "Deep feature pyramid reconfiguration for object detection," in *Proceedings of the European Conference on Computer Vision*, 2018, pp.169-185.

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

[24] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp.7132-7141.

[25] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp.510-519.

[26] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM:Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision*, 2018, pp.3-19.

[27] Y.L. Li, S,J Wang, "HAR-Net:Joint learning of hybrid attention for single-stage object detection," *arXiv preprint arXiv:1904.11141*, 2019.

[28] Ioffe, Sergey, and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[29] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, 2010, pp.303-338.

[30] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, "Microsoft COCO: Common objects in context," in *European conference on computer vision*, 2015, pp.740-755.

[31] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, "Automatic differentiation in PyTorch," *31st Conference on Neural Information Processing Systems*, 2017.

[32] K. He, X. Zhang, S. Ren and J. Sun, "Delving deep into Rectifiers: Surpassing human-level performance on ImageNet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026-1034.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp.770-77, .

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

[35] Z. Zhang, S. Qiao, C. Xie, W. Shen, B. Wang, and A. L. Yuille, "Single-Shot object detection with enriched semantics," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp.5813-5821.

[36] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-Outside Net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp.2874-2883.

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

[38] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp.6848-6856.

[39] J. Nie, R. M. Anwer, H. Cholakkal, F. S. Khan, Y. Pang, and L. Shao, "Enriched feature guided refinement network for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp.9537-9546.

[40] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp.1440-1448.

[41] T. Y. Lin, P. Goyal, R. Girshick , K. He and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980-2988.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE