FAS-Net: Construct Effective Features Adaptively for Multi-Scale Object Detection

Jiangqiao Yan,[1,2,3] **Yue Zhang,**[1] **Zhonghan Chang,**[1] **Tengfei Zhang,**[1]
Menglong Yan,[1] **Wenhui Diao,**[1] **Hongqi Wang,**[1] **Xian Sun**[1*]
[1]Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China
[2]Key Laboratory of Network Information System Technology(NIST),
Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China
[3]School of Electronic, Electrical and Communication Engineering,
University of Chinese Academy of Sciences, Beijing 100190, China
yanjiangqiao16@mails.ucas.edu.cn, zhangyuereal@163.com, {changzhonghan16, zhangtengfei16}@mails.ucas.ac.cn,
yanmenglong@foxmail.com, Wiecas@sina.com, {whdiao, sunxian}@mail.ie.ac.cn

Abstract

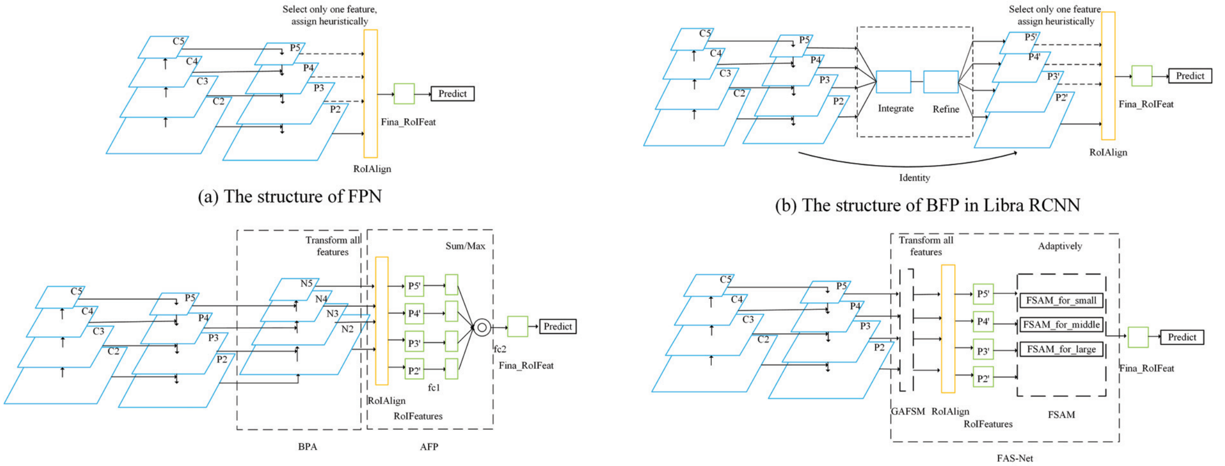
Feature pyramid is the mainstream method for multi-scale object detection. In most detectors with feature pyramid, each proposal is predicted based on feature grids pooled from only one feature level, which is assigned heuristically. Recent studies report that the feature representation extracted using this method is sub-optimal, since they ignore the valid information exists on other unselected layers of the feature pyramid. To address this issue, researchers present to fuse valid information across all feature levels. However, these methods can be further improved: the feature fusion strategies, which use common operation (element-wise max or sum) in most detectors, should be replaced by a more flexible way. In this work, a novel method called feature adaptive selection subnetwork (FAS-Net) is proposed to construct effective features for detecting objects of different scales. Particularly, its adaption consists of two level: global attention and local adaptive selection. First, we model the global context of each feature map with global attention based feature selection module (GAFSM), which can strengthen the effective features across each layer adaptively. Then we extract the features of each region of interest (RoI) on the entire feature pyramid to construct a RoI feature pyramid. Finally, the RoI feature pyramid is sent to the feature adaptive selection module (FASM) to integrate the strengthened features according to the input adaptively. Our FAS-Net can be easily extended to other two-stage object detectors with feature pyramid, and supports to analyze the importance of different feature levels for multi-scale objects quantitatively. Besides, FAS-Net can also be further applied to instance segmentation task and get consistent improvements. Experiments on PASCAL07/12 and MSCOCO17 demonstrate the effectiveness and generalization of the proposed method.

Introduction

Large scale variation across objects is one of the main factors affecting the performance of the object detectors. To alleviate this problem, multiple solutions have been proposed. A classical strategy to address this issue is to run the detector over a number of scaled input images, which is called featurized image pyramids (Lin et al. 2017). But this method increases the inference time and memory usage of the detector significantly. In contrast, considering the inherent hierarchical structure of convolutional neural networks (CNN), algorithms using multiple CNN layers have been proposed in recent years (Liu et al. 2018a). As the low-level and high-level information are complementary for object detection, the feature pyramid structure is proposed (Lin et al. 2017; Kong et al. 2017; Zhang et al. 2018) and becomes the mainstream method to solve the multi-scale object detection problem. Among them, FPN (Lin et al. 2017) is the most representative method. It integrates features via lateral connections in the sequential manner and each proposal is predicted based on feature grids pooled from one feature level, which is assigned heuristically (Fig. 1(a)).

Although achieving encouraging results, FPN still has some limitations. For example, Libra R-CNN (Pang et al. 2019) proves that feature integration via lateral connections in the sequential manner will make the integrated features focus more on adjacent resolution but less on others. This means that some useful semantic information in highest-level can not transfer to the lowest-level features effectively. They propose the balanced feature pyramid (BFP) structure to resolve the feature level imbalance (Fig. 1(b)). PANet (Liu et al. 2018b) finds it is not optimal that each proposal is predicted from a single heuristically assigned feature level. In PANet, the adaptive feature pooling (AFP) is proposed to use all levels feature maps for each proposal and fusing them for following prediction (Fig. 1(c)). M2Det (Zhao et al. 2019) points out that objects of the same size with different apparent complexity also need to extract valid information from different levels of feature layers. Thus, each feature map (used for detecting objects in a specific range of size in FPN) in the pyramid mainly or only consists of single-level features will result in suboptimal detection performance. In order to solve the shortcomings of the general FPN structure, they propose multi-level feature pyramid network (MLFPN) to construct more effective feature pyramids for detecting objects of different scales.

[*]Corresponding author.



(a) The structure of FPN

(b) The structure of BFP in Libra RCNN

(c) The structure of BPA and AFP in PANet

(d) The structure of FPN with FAS-Net

Figure 1: (a)The structure of FPN (b) The structure of BFP in Libra RCNN (c) The structure of BPA and AFP in PANet (d) The structure of FPN with FAS-Net

However, the above methods still have some problems. For M2Det, the MLFPN introduce too many parameters and ignore the valid information in the low-level features. For Libra-RCNN and PANet, although they try to fuse valid information across all feature levels, the feature fusion operation in their structure (such as BFP in Libra RCNN or AFP in PANet) can only occupy a limited part of the feature fusion space, which can be further optimized by a more flexible way. In this paper, we propose a novel feature adaptive selection subnetwork (FAS-Net, Fig. 1(d)) to construct more effective feature representations for objects of different scales, globally and locally. FAS-Net consists of: (1) a global attention based feature selection module (GAFSM) and (2) a group of feature adaptive selection modules (FASM). Firstly, we use the GAFSM to strengthen effective feature channels based on the global context. Then enhanced features are sent to FASM to get a combined feature representation adaptively over all feature maps for each RoI. Extensive experiments demonstrate the effectiveness and generality of the proposed method.

To sum up, the main contributions of this paper can be summarized as follows: (1) FAS-Net is proposed. It helps to construct effective feature representation for multi-scale objects, globally and locally. (2) quantitative analysis reveals that each feature level in the feature pyramid has different, as well as positive effects on objects of different scales. (3) extensive implementations with various two-stage detectors based on FPN structure prove the generality and robustness of FAS-Net. Furthermore, FAS-Net is plugged into the Mask RCNN (He et al. 2017) for instance segmentation and achieves consistent performance improvement on objects of all scales.

Related Works

There are two strategies to alleviate the problems arising from objects with various scales. The featurized image pyramid is used in the era of hand-engineered features. Although it is optimized by training strategy such as SNIP (Singh and Davis 2018) and SNIPER (Singh, Najibi, and Davis 2018), such a method is still time-wise and memory-wise, which forbid its application in most practical work. In contrast, the feature pyramid structure, such as FPN, use a top-down architecture with lateral connections to build high-level semantic feature maps at all scales. It shows significant improvement as a generic feature extractor and can be easily integrated into the state-of-the-art CNN based detectors, yielding an end-to-end solution.

However, some recent work indicates that there are still some problems in the FPN structure. In FSAF (Zhu, He, and Savvides 2019), researchers point out the problem of heuristic-guided feature selection used in FPN. And they propose the online feature selection to dynamically select the most suitable level of feature for each instance during training. M2Det (Zhao et al. 2019) find that each feature map in the pyramid mainly or only consists of single-level feature, which will result in suboptimal detection performance. Multi-level feature pyramid network (MLFPN) is proposed to construct a more effective feature pyramid. But the MLFPN increase the complexity of the model significantly. And in FSAF and M2Det, the feature representation of each RoI is extract from a single feature level, which ignore the valid information exists on other feature levels of the pyramid. In contrast, Libra RCNN and PANet show that all feature map layers contain valid information for object detection regardless of the size of the object. They use the common operation (element-wise max or sum) to fuse valid information from all feature levels.

Furthermore, the update of network topology in recent years has also significantly improved the performance of detectors. The SENet (Hu, Shen, and Sun 2018) and GCNet (Cao et al. 2019) strengthen the original features by the same features aggregated from all positions. They can effectively model channel-wise feature dependencies or long-range dependencies. In the field of object classification, SKNet (Li et al. 2019a) is proposed to allow each neuron in the CNN to adaptively adjust its receptive field size based on multiple scales of input information. All these methods show that features can be selected or recalibrated adaptively by using modules of specific structures. And similar with SKNet, TridentNet (Li et al. 2019b) points out that objects of different scales should have some differences in effective feature extraction.

Proposed Method

We design our structure following the principles below: (1) extract valid feature representations from all feature map layers (2) design a new module that can get a more flexible fusion features based on the input content to replace the rigid fusion method which is set manually (3) to make objects of different scales get more effective feature representations, we use different branches (same structure but with different parameters) to fuse effective features in the pyramid. As shown in Fig. 1(d), FAS-Net mainly consists of two modules: a group of feature adaptive selection modules (FASM) and the global attention based feature selection module (GAFSM). Before we extract features for all proposals, we use the GAFSM to enhance the feature maps in feature pyramid according to the global attention. Then we choose a specific FASM based on the size of the proposals. The FASM can output the final RoI features which fuse all valid information adaptively. More details about the two core modules and network configurations of FAS-Net are introduced in the following.

GAFSM

Given the feature hierarchy, the aim of the GAFSM is to enhance informative features and suppress less useful ones globally for each specific scale. As shown in SENet (Hu, Shen, and Sun 2018) and GCNet (Cao et al. 2019), the introduction of global context information in the network can further improve the discriminability of the features. But the SE block and the GC block mainly used in the backbone network, which neglect the output features of FPN. When the feature levels of different depths are enhanced by the FPN structure, there may be some change in the importance of each feature channel. Therefore, it is necessary to further re-construct the global attention information to process all output layers of the FPN. We add GAFSM to strengthen the informative features on each level of FPN.

In this paper, we apply the simplified non-local (SNL) block (Cao et al. 2019) as the basic module. SNL can be replaced by GC block to reduce the amount of parameters. In order to remove the impact of additional normalized layers in GC block, we use the SNL modules here to ensure that performance gains are primarily affected by global context attention. SNL is a simplified implementation of the NL (Wang et al. 2018) module. As the global contexts modeled by non-local network are almost the same for different query positions within an image, we can simplify the NL module as follows:

$$\mathbf{z}_i = \mathbf{x}_i + \mathbf{W}_v \sum_{j=1}^{N_p} \frac{\exp(\mathbf{W}_k \mathbf{x}_j)}{\sum_{m=1}^{N_p} \exp(\mathbf{W}_k \mathbf{x}_m)} \mathbf{x}_j \tag{1}$$

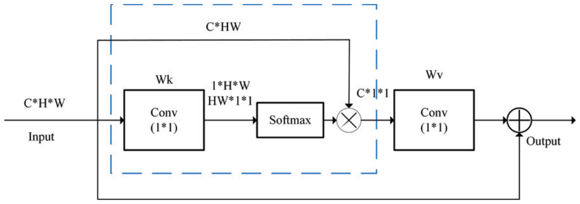


Figure 2: The structure of SNL, the context modeling of structure is shown in blue dashed boxes

where i is the index of query positions and j enumerates all possible positions in the input. $\mathbf{W}_k$ and $\mathbf{W}_v$ denote linear transform matrices, we use 1×1 in our module. We directly model global context as a weighted average of the features at all positions, and add the global context features to the features at each query position. The $\mathbf{W}_v$ is used to recalibrate the importance of channels.

The structure of SNL is shown in Fig. 2. The original SNL module is developed to strengthen the output features of the backbone network. In contrast, the SNL in our GAFSM is used to construct a global attention map to process each feature layer of the feature pyramid, which is only used before extracting RoI features by ROIAlign.

FASM

Inspired by the dynamic selection mechanism in SKNet, we design our FASM to construct fused features adaptively from all feature maps in feature pyramid. The structure of FASM is illustrated in Fig. 3, which can be divided into three parts: fuse, gated weight calculation and feature integrated. As shown in Fig. 1(d), we map each proposal to different feature levels in feature pyramid, and following Mask RCNN, RoIAlign is used to pool feature grids from each level to get the RoI feature pyramid $\{\mathbf{P2}^{'}, \mathbf{P3}^{'}, \mathbf{P4}^{'}, \mathbf{P5}^{'}\}$. As default setting in FPN, the dimensions of RoI features is $n \times C \times 7 \times 7$ in detection network, where n is the number of RoIs send into the FASM and C is the number of channels, which is set to 256 by default. Then the RoI feature pyramid send into the FASM to adaptively select valid features according to the input content.

Fuse: We first fuse information from all branches via an element-wise summation to get the Feature_Sum $\mathbf{U}$:

$$\mathbf{U} = \mathbf{P2}^{'} + \mathbf{P3}^{'} + \mathbf{P4}^{'} + \mathbf{P5}^{'} \tag{2}$$

and we use global average pooling to generate channel-wise statistics as $\mathbf{s} \in \mathbb{R}^C$. Specifically, the c-th element of $\mathbf{s}$ is calculated by shrinking $\mathbf{U}$ through spatial dimensions 7×7:

$$\begin{aligned} \mathbf{s}_c &= F_{gp}(\mathbf{U}) \\ &= \tfrac{1}{49} \sum_{i=1}^{7} \sum_{j=1}^{7} \mathbf{U}_c(i, j) \end{aligned} \tag{3}$$

then we transform the $\mathbf{s}$ with a simple fully connected (fc) layer to enable the guidance for the precise and adaptive selections. Similar to the SE block in SENet, we use a bottleneck form to reduce the number of parameters from $C \times C$ to $C \times C/r$, where r is the bottleneck ratio and C/r denotes

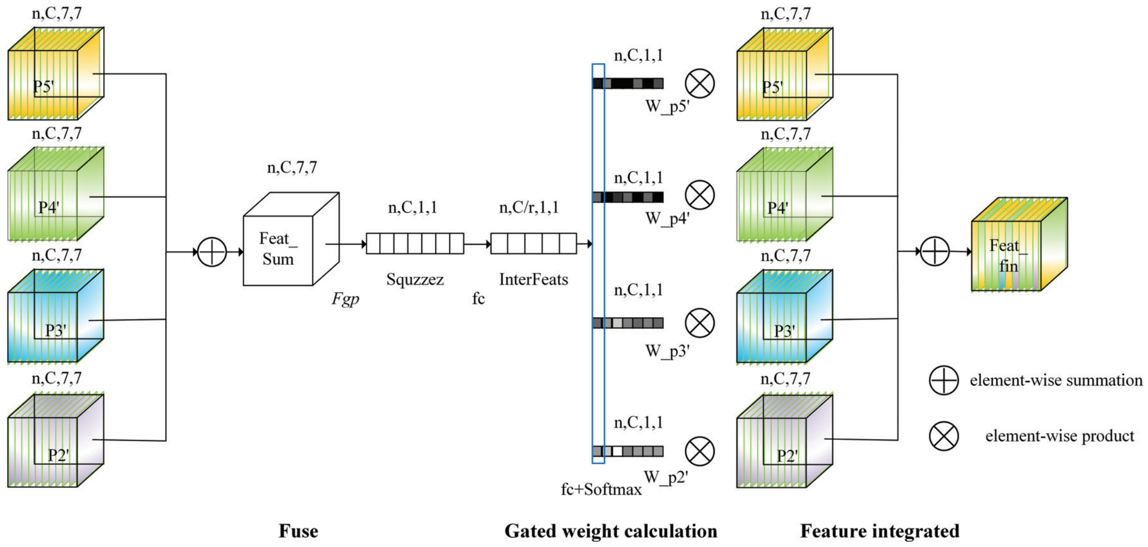


Figure 3: The structure of FASM

the hidden representation dimension of the bottleneck. The inter-feature $\mathbf{z}$ is calculated as follows:

$$\mathbf{z} = F_{fc}(\mathbf{s}) = \delta(\mathbf{W}\mathbf{s}) \tag{4}$$

where $\mathbf{z} \in \mathbb{R}^{d\times 1}$, $d = C/r$, δ is the ReLU function (Nair and Hinton 2010), $\mathbf{W} \in \mathbb{R}^{d\times C}$. Compared with selective kernel (SK) unit, we remove the batch normalization (Ioffe and Szegedy 2015) in this operation because of the small batch size during our training process (batch size is set to 1 for experiments on VOC and 2 for experiments on COCO).

Gated weight calculation: After the fuse stage, we calculate weights for each feature in the RoI feature pyramid. We use another fc layer to transform the inter-feature and followed by a softmax operator, which is applied on the channel-wise digits:

$$\begin{aligned}
\mathbf{W\_P2}^{'}_{c} &= \frac{e^{\mathbf{A}_c\mathbf{z}}}{e^{\mathbf{A}_c\mathbf{z}}+e^{\mathbf{B}_c\mathbf{z}}+e^{\mathbf{D}_c\mathbf{z}}+e^{\mathbf{E}_c\mathbf{z}}}\\
\mathbf{W\_P3}^{'}_{c} &= \frac{e^{\mathbf{B}_c\mathbf{z}}}{e^{\mathbf{A}_c\mathbf{z}}+e^{\mathbf{B}_c\mathbf{z}}+e^{\mathbf{D}_c\mathbf{z}}+e^{\mathbf{E}_c\mathbf{z}}}\\
\mathbf{W\_P4}^{'}_{c} &= \frac{e^{\mathbf{D}_c\mathbf{z}}}{e^{\mathbf{A}_c\mathbf{z}}+e^{\mathbf{B}_c\mathbf{z}}+e^{\mathbf{D}_c\mathbf{z}}+e^{\mathbf{E}_c\mathbf{z}}}\\
\mathbf{W\_P5}^{'}_{c} &= \frac{e^{\mathbf{E}_c\mathbf{z}}}{e^{\mathbf{A}_c\mathbf{z}}+e^{\mathbf{B}_c\mathbf{z}}+e^{\mathbf{D}_c\mathbf{z}}+e^{\mathbf{E}_c\mathbf{z}}}
\end{aligned} \tag{5}$$

where $\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{E} \in \mathbb{R}^{C\times d}$ denote linear transform matrices that can map the inter-feature $\mathbf{z}$ to C channel. Each of the channel is corresponding to the channel of the input feature (e.g. $\mathbf{P2}^{'}$). Then we can concatenate all transformed features (total 4C dimensions) into a softmax layer to calculate the weight values of each feature layer on a specific channel c. $\mathbf{W\_P2}^{'}$, $\mathbf{W\_P3}^{'}$, $\mathbf{W\_P4}^{'}$, $\mathbf{W\_P5}^{'}$ denote the soft attention vector for each feature in RoI feature pyramid.

Feature integrated: Finally, we get an adaptive weighted combination feature for each proposal. The final feature map $\mathbf{V}$ is obtained through the attention weights on various features:

$$\mathbf{V}_c = \sum_{i=2}^{5} \mathbf{W\_Pi}^{'}_{c} \cdot \mathbf{Pi}^{'}_{c}, \sum_{i=2}^{5} \mathbf{W\_Pi}^{'}_{c} = 1 \tag{6}$$

where $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_C], \mathbf{V}_c \in R^{7\times 7}$.

The original SK unit is developed for explicitly select features between all branches with different kernel sizes, and shows great success in object classification. In contrast, we apply our FASM to fuse valid information from hierarchy features. FASM helps to boost feature discriminability and select more useful information based on specific local input. In addition, we have more than one FASM in our FAS-Net. According to the size of the proposals, we divide them into three parts and send them to different FASMs. Our definition of large, medium and small proposals is consistent with the COCO data set. When the area of the proposal mapped back to the original image is less than 32 × 32, we define it as a small one. All the features of small proposals should be sent to FASM_for_small branch. When the area of the proposal mapped back to the original image is larger than 96 × 96, we define it as a large one. All the features of large proposals should be sent to FASM_for_large branch. In the **Ablation Experiments**, the impact of the number of FASMs will be discussed in detail.

Network Configurations

The proposed FAS-Net can be integrated into various two-stage detectors with FPN structure, like Faster RCNN (Ren et al. 2015), Libra RCNN, DCN (Dai et al. 2017; Zhu et al. 2018) and GCNet. Before training the whole network, the backbone network should be pre-trained on the ImageNet 2012 dataset (Russakovsky et al. 2015). All the default configurations of FAS-Net contain 4 SNL in GAFSM and 3 FASM. As for input size, we follow the settings in mmdetection[1], when training on the Pascal VOC dataset (Everingham et al. 2010), the input images resize to (1000,600), when training on the COCO dataset (Lin et al. 2014), the input images resize to (1333,800). The first number in paren-

[1] https://github.com/open-mmlab/mmdetection

theses indicates the maximum value of the width or height of the image after the scale change, and the second number indicates the minimum value. The image after the scale change will be padded so that the width and height of the input feature can be divisible by 32.

Experiments

We conduct experiments on three widely used datasets: PASCAL VOC 2007, PASCAL VOC 2012 and MS COCO 2017. All network backbones are pretrained on the ImageNet1k classification set and fine-tuned on the detection dataset. We use the pre-trained ResNet-50 that is publicly available. Our experiments are based on re-implementation of various modern detectors using PyTorch[2] and mmdetection. All hyper-parameters follow the settings in mmdetection if not specifically noted. We first evaluate the proposed methods with ablation experiments. Then we apply our module for various modern detectors to verify its effectiveness and generality. For evaluation measure, we use average precision (AP) per class which is a standard metric for object detection. It is evaluated by computing the area under the precision-recall curve. We also compute mean average precision (mAP) by averaging the APs over all object classes. Furthermore, for experiments on the COCO dataset, we report results follow standard COCO-style average precision metrics that include AP (averaged over IoU thresholds), AP_{50} (AP for IoU threshold 0.5), AP_{75} (AP for IoU threshold 0.75). We also include AP_S, AP_M, AP_L, which correspond to the results on small, medium and large scales respectively.

Ablation Experiments

To show the effectiveness of the FASM and the GAFSM, we perform several ablation studies. For this evaluation, we train detectors on the VOC2007 and VOC2012 trainval set and test them on the VOC2007 test set. All experiments on VOC dataset are trained with a single RTX 2080 GPU, CUDA 10 and cuDNN 7, without parallel and distributed training. We initialize the learning rate as 1×10^{-3}, and then decrease it to 1×10^{-4} at 9 epochs, and stop at 12 epochs. The batch size is set to 1 to remove the impact of Batch Normalization on network performance. We use ResNet50 FPN Faster-RCNN as the baseline.

Effects of proposed methods: In Table 1, we first show mAPs of the baseline detector without the FASM and GAFSM. Then we improve the detector by using each of them or the overall FAS-Net. In order to analyze the impact of the proposed structure on multi-scale object detection problems, we list the detection performance of the network on objects of different scales, just like the AP_S, AP_M, AP_L in COCO dataset. Firstly, we extend the baseline detector with our FASM and the AP improves from 80.48 to 81.66 as illustrated in the third row of Table 1. It demonstrates that our FASM can adaptively select effective information from each feature map of the feature pyramid. And for objects of different scales, using FASM can effectively enhance the discriminativeness of the feature representation. Then we extend the baseline detector with GAFSM and the AP improves from 80.48 to 81.47 as illustrated in the fourth row. The global attention makes the network enhance the features with suitable semantics and helps detect objects with scale variation. Finally, we extend the baseline detector with FAS-Net, and the AP improves from 80.48 to 82.18, which indicates the complementarity between the proposed two modules.

Table 1: Ablation study on PASCAL VOC: effects of the proposed FASM and the GAFSM.

detectors	AP_S	AP_M	AP_L	mAP
baseline (Lin et al. 2017)	43.88	66.27	84.29	80.48
baseline+FASM	47.11	68.82	84.88	81.66
baseline+GAFSM	45.91	**69.85**	84.53	81.47
baseline+FAS-Net	**47.46**	69.39	**85.17**	**82.18**

Table 2: Effects of the number of FASM.

Number of FASM	AP_S	AP_M	AP_L	mAP
0	43.88	66.27	84.29	80.48
1	45.02	67.28	84.47	80.70
3	**47.11**	**68.82**	84.88	**81.66**
4	44.24	68.68	**84.94**	81.53

The number of the FASM: As we can use different FASM branches in the network to handle objects in different scales, we investigate on the number of FASM we should use in our network. In theory, using a single FASM can learn the feature weight distribution for all objects. However, we find that it is very difficult to learn the feature weight distribution for objects in all scales simultaneously with only one FASM, especially when the training time is limited. Inspired by TridentNet (Li et al. 2019b), which construct a parallel multi-branch architecture and specialize each branch by sampling object instances of proper scales for training. We use more than one FASM in our FAS-Net and specialize each FASM branch by sampling object instances of proper scales for training. As shown in Table 2, we extend the baseline detector with different number of FASM, where 0 corresponds to the baseline network. The number of FASM is set to 3 by default, the scale division point is set to 32^2 and 96^2. When the number of FASM is 4, we set the scale division point to 32^2, 96^2 and 160^2. The detection performance is related to the number of FASMs and we get the best results when the number of the FASM is set to 3. When using more FASMs (more than 3), in order to learn the weight distribution of additional branches, it may have a large impact on the features extracted by the backbone framework, thus affecting the effective detection of targets in other scales. As shown in the last row of Table 2, subdividing large objects will affect the detection performance of small objects.

Quantitative analysis of the weight: In order to analyze the importance of different feature levels for multi-scale objects quantitatively, we calculate the weights on each feature map in RoI feature pyramid learned by the FASM. When the test image is sent to the network, the weight assigned to each channel of the specific feature map is learned by a FASM

[2] https://pytorch.org/

Table 3: Changes in weight distribution learned by FASM.

Number of FASM	branch	weight for P2’	weight for P3’	weight for P4’	weight for P5’
1	small_proposals	0.2070	0.2343	0.2421	0.3166
	middle_proposals	0.1949	0.2267	0.2422	0.3362
	large_proposals	0.1835	0.2156	0.2372	0.3637
3	FASM_for_small	0.2683	0.2785	0.2387	0.2144
	FASM_for_middle	0.2209	0.2533	0.2622	0.2636
	FASM_for_large	0.1903	0.2122	0.2212	0.3763

branch. Then the mean value of the learnd weights is shown in Table 3. For example, the weight for $\mathbf{Pi}^{'}$ is calculated as follows:

$$weight\_for\_pi^{'} = \frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C}\mathbf{W\_Pi}^{'}_{n,c} \qquad (7)$$

where N denotes the number of objects in a specific size range.

The feature layer with larger weight means that more valid information is extracted from this layer. As shown in the last row of Table 3, for large objects, an average of 37.63% of the valid information comes from the highest feature level $\mathbf{P5}^{'}$. Consistent with our previous experience, as the size of proposals increases, more effective information comes from high-level features. For small objects, more effective information comes from low-level features. In addition, all the feature layers contains valid information for detection regardless of the size of the objects. Comparing the weights learned when using different number of FASMs, we can see that although a single FASM can be used to learn the correct trend (the weight for $\mathbf{P2}^{'}$ increase from 18.35% to 20.7% when the input objects become smaller), the final weight assignment value is not effective enough compared with using multiple FASMs. As shown in the fifth row, for small objects, the weight for $\mathbf{P2}^{'}$ should be larger than the weight for $\mathbf{P5}^{'}$, which is hard to achieve for a single FASM as the number of small objects in the data set is much smaller than the number objects in other scale.

The reduction ratio in the FASM: Reduction ratio r introduced in FASM is a hyperparameter which allows us to vary the capacity and computation cost of the FASM. To investigate the trade-off between performance and computational cost modified by this hyperparameter, we conduct experiments with ResNet50 FPN Faster-RCNN extend by our FASM for a range of different r values. The comparison in Table 4 shows that performance improve monotonically with increased complexity. We found that setting r=8 achieve a good balance between accuracy and complexity. We set r=8 for all experiments reported in this work.

Table 4: The effect of reduction ratio in FASM

Reduction Ratio	AP$_S$	AP$_M$	AP$_L$	mAP
4	48.13	69.43	84.95	81.84
8	47.11	68.82	84.88	81.66
16	44.52	68.67	84.98	81.27

Effectiveness and generality on PASCAL VOC

To prove the effectiveness and generality of FAS-Net, we test extensive implementations of FAS-Net with various two-stage detectors based on FPN structure. ResNet50 is used as backbone feature extractor in all implementations. As shown in Table 5, besides the basic FPN, detectors including GCNet, DCN and Libra RCNN all achieve performance improvement with FAS-Net. The AP, AP$_S$, AP$_M$, AP$_L$ all improve consistently, especially the AP$_S$ improves by 3.9% on average. It is noteworthy that, although GCNet uses global context information to enhance the effective features in the backbone network, the combination of our FAS-Net can further improve the network detection performance. This shows that our modules are complementary to algorithms that modeling and using global context information to enhance features in backbone network. For Libra RCNN, considering that not all modules are related to our proposed structure, we only use their BFP module for comparison experiments. As shown in the last two rows of Table 5, although the global attention information is used and the semantic information of each layer is balanced by using BFP in Libra RCNN, we can get more efficient feature representation by using our FAS-Net structure.

Similar to our methods, PANet also try to fuse valid information from all the feature maps by using the bottom-up path augmentation (BPA) and the adaptive feature pooling (AFP) they proposed. For fair comparison, we use the same environment and parameter settings, and extend the FPN ResNet50 Faster RCNN detector using BPA and AFP or our FAS-Net. As shown in the second and the last row of Table 5, our method achieves a better performance with a lower complexity, which demonstrate the effectiveness of our FAS-Net. It can be shown that our performance improvement is not due to the increase of the complexity of the model as we achieve a better performance with a smaller model size.

Effectiveness on MS COCO 2017

To further validate the proposed framework on a larger and more challenging dataset, we conduct experiments on MS COCO 2017. we train all detectors with COCO_2017_train set and test them on the COCO_2017_val set as done in mmdetection. All experiments using COCO dataset are trained with two RTX 2080 GPUs with distributed train-

Table 5: Comparative experimental results on VOC datasets.

Detector	Model Size(KB)	AP$_S$	AP$_M$	AP$_L$	AP	FPS
FPN Faster RCNN (Lin et al. 2017)	161617	43.88	66.27	84.29	80.48	22.5
FPN + FAS-Net	163132	47.46	69.39	85.17	82.18	17.8
GCNet (Cao et al. 2019)	171484	49.27	68.24	84.66	81.74	21.1
GCNet+FAS-Net	173000	52.68	69.97	85.58	82.21	16.6
DCN (Zhu et al. 2018)	165024	41.08	68.71	86.17	82.85	20.8
DCN+FAS-Net	166539	47.32	70.47	86.51	83.46	16.9
Libra RCNN	162646	49.91	68.29	84.81	81.62	21.9
Libra RCNN+FAS-Net	164162	52.38	69.88	85.10	82.10	17.0
PANet (Liu et al. 2018b)	175449	45.70	68.26	84.73	81.39	17.1

Table 6: Detection results on COCO datasets

Module	bbox, Avg. Precision, IoU			bbox, Avg. Precision, Aera		
Name	AP@0.5:0.95	AP@0.5	AP@0.75	AP$_S$	AP$_M$	AP$_L$
FPN Faster RCNN (Lin et al. 2017)	35.75	57.12	38.52	20.54	39.31	45.67
FPN+FAS-Net	36.75	59.14	39.22	22.19	40.69	46.12
Mask RCNN (He et al. 2017)	36.63	57.84	39.53	21.35	40.02	47.30
Mask RCNN+FAS-Net	37.39	59.52	40.09	22.17	41.49	47.69

Table 7: Segmentation results on COCO datasets

Module	segm, Avg. Precision, IoU			segm, Avg. Precision, Aera		
Name	AP@0.5:0.95	AP@0.5	AP@0.75	AP$_S$	AP$_M$	AP$_L$
Mask RCNN (He et al. 2017)	33.62	54.53	35.72	17.70	36.58	45.82
Mask RCNN+FAS-Net	34.69	56.27	36.82	18.34	38.41	47.13

ing. we initialize the learning rate as 5×10^{-3}, and then decrease it at 8 and 11 epochs by a factor of 0.1, and stop at 12 epochs. The batch size is set to 2 (one image on each GPU). As shown in Table 6, our module can effectively improves the performance of object detection task. Our FAS-Net can achieve consistent improvement on all evaluation metrics, especially on the AP@0.5, we get a 1.8% improvement on average.

Generality to Instance Segmentation

We find that our FASM can also be embedded into the mask branch of the Mask RCNN. Thus, we extend the Mask RCNN framework with our FAS-Net to learn instance segmentation task. As shown in Table 7, our module can effectively improves the performance of instance segmentation task. We achieve an average 1% increase on all evaluation metrics. We believe that the proposed method can be applied to similar network structures to further solve other computer vision problems that require effective feature representation. With the introduction of some low-level features by our FASM module for large objects, we get better segmentation results at the details of large objects (Fig. 4).

Discussion

In this paper, the scale range of proposals corresponding to different FASMs is set manually. We think we can further improve the final performance of the network by selecting a better scale division strategy based on the specific data set.

Furthermore, it should be noted that our FASM module is effective under the condition that the features of the same

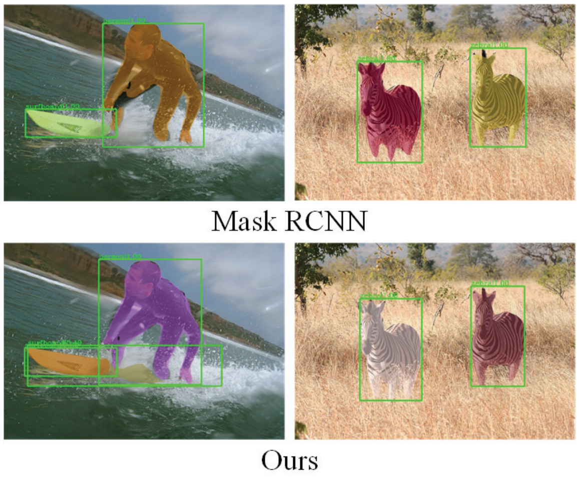


Figure 4: The segmentation results on COCO

spatial location on different feature levels represent the same content. When the feature representation of the same spatial location on different feature maps are inconsistent, we cannot obtain effective combined features by using FASM. For example, we can not use our FAS-Net on GA-RPN (Wang et al. 2019) based detectors because of the anchor-guided feature adaptation module, which will cause inconsistencies in the representation of the same location on different feature maps.

Conclusion

In this work, a novel method called feature adaptive selection subnetwork (FAS-Net) is proposed to construct effective features for detecting objects of different scales. FAS-Net consists of two novel modules: GAFSM and FASM. GAFSM can strengthen the effective features in each layer according to the global attention. The FASM can integrate the effective features according to the input adaptively and can quantify the weight, which effectively proves the complementarity of the effective information on each feature map of the feature pyramid. A two-stage object detector with FPN structure can be easily extend by our FAS-Net and get an efficient performance improvement. The structure can be further applied to other visual fields, such as instance segmentation. Adequate experiments demonstrate the effectiveness and the generality of the proposed architecture and the novel modules.

Acknowledgments

The work is supported by the National Natural Science Foundation of China under Grants 41801349.

References

Cao, Y.; Xu, J.; Lin, S.; Wei, F.; and Hu, H. 2019. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. *arXiv: Computer Vision and Pattern Recognition*.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. *international conference on computer vision* 764–773.

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision* 88(2):303–338.

He, K.; Gkioxari, G.; Dollar, P.; and Girshick, R. B. 2017. Mask r-cnn. *international conference on computer vision* 2980–2988.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. *computer vision and pattern recognition* 7132–7141.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *international conference on machine learning* 448–456.

Kong, T.; Sun, F.; Yao, A.; Liu, H.; Lu, M.; and Chen, Y. 2017. Ron: Reverse connection with objectness prior networks for object detection. *computer vision and pattern recognition* 5244–5252.

Li, X.; Wang, W.; Hu, X.; and Yang, J. 2019a. Selective kernel networks. *arXiv: Computer Vision and Pattern Recognition*.

Li, Y.; Chen, Y.; Wang, N.; and Zhang, Z. 2019b. Scale-aware trident networks for object detection. *arXiv: Computer Vision and Pattern Recognition*.

Lin, T.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. *european conference on computer vision* 740–755.

Lin, T.; Dollar, P.; Girshick, R. B.; He, K.; Hariharan, B.; and Belongie, S. J. 2017. Feature pyramid networks for object detection. *computer vision and pattern recognition* 936–944.

Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P. W.; Chen, J.; Liu, X.; and Pietikainen, M. 2018a. Deep learning for generic object detection: A survey. *arXiv: Computer Vision and Pattern Recognition*.

Liu, S.; Qi, L.; Qin, H.; Shi, J.; and Jia, J. 2018b. Path aggregation network for instance segmentation. *computer vision and pattern recognition* 8759–8768.

Nair, V., and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. 807–814.

Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; and Lin, D. 2019. Libra r-cnn: Towards balanced learning for object detection. *arXiv: Computer Vision and Pattern Recognition*.

Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2015. Faster r-cnn: towards real-time object detection with region proposal networks. 2015:91–99.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. S.; et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115(3):211–252.

Singh, B., and Davis, L. S. 2018. An analysis of scale invariance in object detection - snip. *computer vision and pattern recognition* 3578–3587.

Singh, B.; Najibi, M.; and Davis, L. S. 2018. Sniper: Efficient multi-scale training. *neural information processing systems* 9310–9320.

Wang, X.; Girshick, R. B.; Gupta, A.; and He, K. 2018. Non-local neural networks. *computer vision and pattern recognition* 7794–7803.

Wang, J.; Chen, K.; Yang, S.; Loy, C. C.; and Lin, D. 2019. Region proposal by guided anchoring. *arXiv: Computer Vision and Pattern Recognition*.

Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; and Li, S. Z. 2018. Single-shot refinement neural network for object detection. *computer vision and pattern recognition* 4203–4212.

Zhao, Q.; Sheng, T.; Wang, Y.; Tang, Z.; Cai, L.; Chen, Y.; and Ling, H. 2019. M2det: A single-shot object detector based on multi-level feature pyramid network. *national conference on artificial intelligence*.

Zhu, X.; Hu, H.; Lin, S.; and Dai, J. 2018. Deformable convnets v2: More deformable, better results. *arXiv: Computer Vision and Pattern Recognition*.

Zhu, C.; He, Y.; and Savvides, M. 2019. Feature selective anchor-free module for single-shot object detection. *arXiv: Computer Vision and Pattern Recognition*.