

Feature Selective Networks for Object Detection

Yao Zhai^{1*}, Jingjing Fu², Yan Lu², and Houqiang Li¹

¹University of Science and Technology of China

²Microsoft Research Asia

zy92918@mail.ustc.edu.cn, {jifu, yanlu}@microsoft.com, lihq@ustc.edu.cn

Abstract

Objects for detection usually have distinct **characteristics in different sub-regions** and different aspect ratios. However, in prevalent two-stage object detection methods, Region-of-Interest (RoI) features are extracted by RoI pooling with little emphasis on these **translation-variant feature components**. We present feature selective networks to reform the feature representations of RoIs by exploiting their disparities among sub-regions and aspect ratios. **Our network produces the sub-region attention bank and aspect ratio attention bank for the whole image.** The RoI-based sub-region attention map and aspect ratio attention map are selectively pooled from the banks, and then used to refine the original RoI features for RoI classification. **Equipped with a light-weight detection subnetwork, our network gets a consistent boost in detection performance** based on general ConvNet backbones (ResNet-101, GoogLeNet and VGG-16). Without bells and whistles, our detectors equipped with ResNet-101 achieve more than 3% mAP improvement compared to counterparts on PASCAL VOC 2007, PASCAL VOC 2012 and MS COCO datasets.

1. Introduction

Recent years have witnessed the rapid advancement of deep neural networks, which yields large performance improvements on image classification and detection tasks. ConvNets [19, 36, 37, 17] designed for image classification have realized impressive representations of image features, outperforming traditional handcrafted features [7, 27]. Object detectors adopting these deep ConvNets improve accuracy significantly on various detection benchmarks [9, 24]. R-CNN [15] firstly uses deep ConvNet to extract proposal region features by initializing parameters from a pre-trained ImageNet [33] classification model. Fast R-CNN [14] develops a Region-of-Interest(RoI) pooling layer to extract

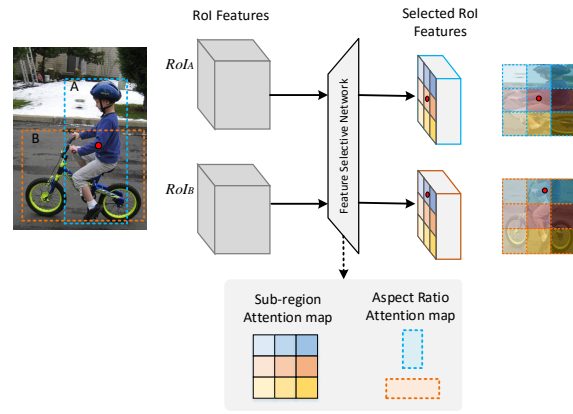


Figure 1. We generate sub-region attention map and aspect ratio attention map for each RoI. Specifically, for the same spatial position (the red point) inside different sub-regions (center in RoI_A , top-center in RoI_B), the sub-region attention map gives different feature attentions. For the RoIs of different aspect ratios, the aspect ratio attention map also produces distinct feature attentions.

RoI features from the convolutional feature maps of the entire image. After that, Faster R-CNN [31] introduces region proposal network (RPN) for generating accurate RoI proposals and sharing computation with detection subnetwork. Although one-stage object detection frameworks [30, 25, 23] have been proposed recently, most of state-of-the-art object detectors adopt **two-stage framework** composed of a proposal generator and a region classifier. Before RoI classification, RoI features are extracted with the following strategy: firstly generate whole image feature maps by ConvNets, and then pool the RoI features.

RoI features matter, as object detection relies on the classification and regression of RoI bounding boxes. Therefore, aside from the progress on deep ConvNets, research has focused on **trying to generate powerful and informative RoI features to boost detection accuracy**. To involve multi-scale features, HyperNet [18] and FPN [22] produce RoI features by utilizing hierarchical feature maps from different depths

*This work was done during an internship at Microsoft Research Asia.

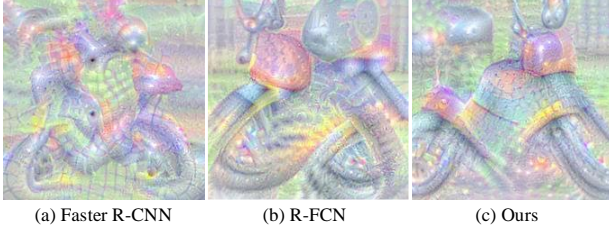


Figure 2. Visualizations of object class “motobike” from the trained models on PASCAL VOC [9] using DeepDraw [29].

of ConvNets. MR-CNN [12] and GBD-net [40] attempt to build a richer region-wise feature representation inside or around the RoI area. All these works adopt the classical RoI pooling layer, which divides RoIs into pooling bins and max-pools the corresponding spatial extent on convolutional feature map into fixed-length RoI features.

RoI features originating from convolutional feature map are high-dimensional (e.g., 2048-d in ResNet-101). To model complex recognition patterns with high-dimensional features, heavy region classification networks are essential, leading to large-scale parameters and time consuming inference. Some might question whether RoI features aiming at region classification and regression actually need so many channels. In general, most of RoIs have already covered a considerable portion of objects thanks to accurate proposal generation [38, 13, 41, 20, 11]. This means that it is possible to highlight the discriminative feature components out of deep features by RoI feature re-extraction.

Moreover, RoI features pooled by classical RoI pooling are *translation-invariant*. Convolutional neural networks *share weights spatially* over all positions on feature maps, forming the translation-invariant feature representations of the whole image. These translation-invariant features are insensitive to detection tasks that focus more on how to precisely localize objects. Notice that objects usually have distinct spatial characteristics in different sub-regions that draw on different feature representations. For example, boundary parts of an object may need more features describing edges and contours for localization, while the center parts favor texture features for classification. Additionally, objects from various categories or viewpoints may also keep varying aspect ratios of ground truth boxes. However, in classical RoI pooling, RoI features are extracted independently for different sub-regions and aspect ratios: features of different sub-regions are pooled on *all channels of convolutional feature map*, with little emphasis on location-related components and aspect ratio preference.

Based on the above observations, we propose *feature selective networks*, which introduce dimension reduction and region-wise *feature attention*. In our network, RoI features are extracted with respect to the sub-region variation and

aspect ratio preference. Benefitting from intense dimension reduction, we replace the multiple high-capacity convolutional or fully connected (*fc*) layers in traditional region classifier by only one low-capacity *fc* layer.

Figure 1 shows a toy example of our feature selective network. Our network produces RoI feature representations with *translation-variant* components based on the detailed sub-region and aspect ratio attention. The channel number C_s (e.g., 40) of RoI features in our network is much smaller than the channel number C of the original convolutional feature map.

Figure 2 shows the DeepDraw visualizations from the trained models of Faster R-CNN, R-FCN [5] and our network. We can observe that our model well maintains **structural characteristics** and preserves the distinct spatial depiction of objects.

The effectiveness of our method is shown through experiments on PASCAL VOC and MS COCO [24]. Our network gets a consistent boost in detection performance based on general ConvNet backbones (ResNet-101 [17], GoogLeNet [37] and VGG-16 [36]).

Our contributions can be summarized into three components:

- Our network generates region-orientated attention maps and creates an informative *translation-variant* representation of RoI features. Without elaborate enhancements, our detector equipped with ResNet-101 achieves more than 3% mAP improvement compared to Faster R-CNN and R-FCN counterparts.
- A heavy detection subnetwork with multiple high-capacity *fc* layers or convolutional layers is simplified to a single low-capacity *fc* layer, which largely reduces parameter size and speeds up inference, especially for ResNet and GoogLeNet backbones.
- Our network is generic and gets a consistent boost based on different ConvNet backbones (ResNet-101, GoogLeNet and VGG-16).

2. Related Work

Traditional Handcrafted Feature Extraction. Deformable Parts Models (DPMs) [10, 1, 8] dominated object detection for years before CNN sprang up. DPM mainly adopts traditional handcrafted features upon all partitioned blocks of candidate boxes with latent SVM classifiers.

Convolutional Feature Extraction. After the success of using deep neural networks for image classification [33], a research stream based on CNNs (OverFeat [34], R-CNN [15]) shows significant improvements in detection accuracy. These methods use convolutional layers to extract features from each region proposal. To further speed up,

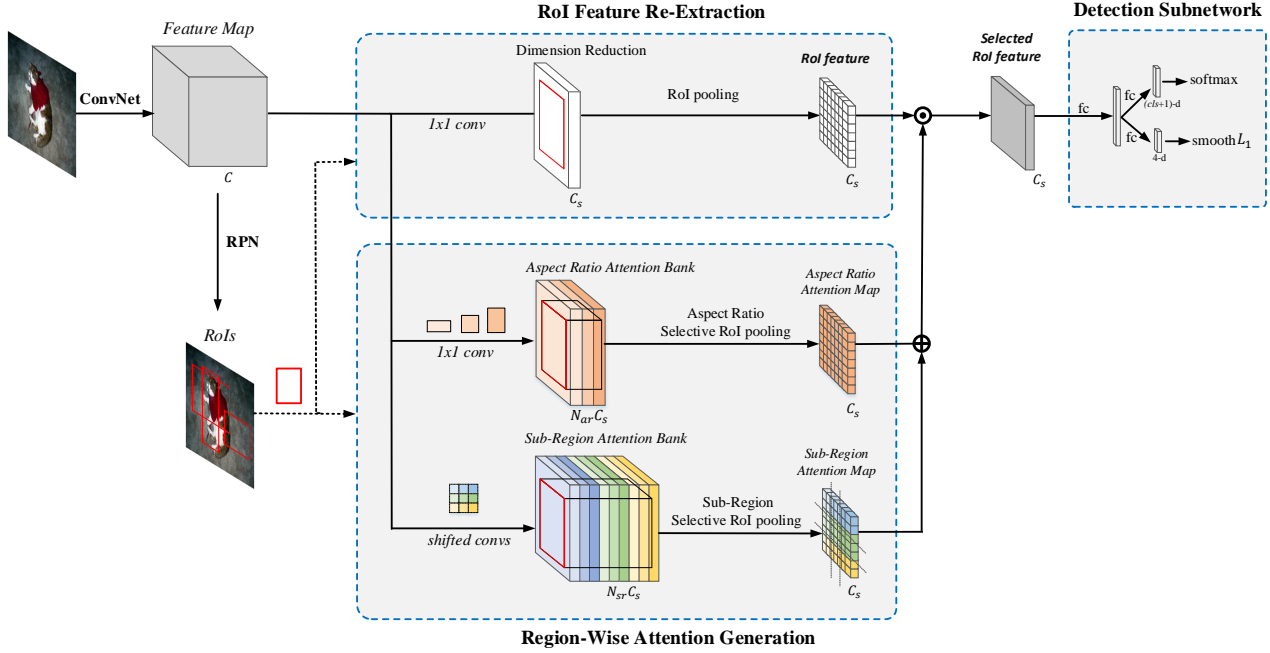


Figure 3. Architecture of our feature selective network. (1) Generate the convolutional feature map of the entire image by ConvNet and produce RoIs by RPN. (2) **RoI Feature Re-Extraction**: Reduce the channel number of feature map from C to C_s (e.g., $C = 2048$, $C_s = 40$) and perform RoI pooling to get the compacted RoI features. (3) **Region-Wise Attention Generation**: Produce $N_{sr}C_s$ -d *sub-region attention bank* and $N_{ar}C_s$ -d *aspect ratio attention bank* according to RoI sub-region division ($N_{sr} = 3 \times 3$) and aspect ratio division ($N_{ar} = 3$), respectively. Then given an RoI, selectively pool the *sub-region attention map* and *aspect ratio attention map* referring its detailed sub-region and aspect ratio information (visualized as different colors in the figure, the pooling size is 7×7). (4) Generate the final selected RoI features by merging the attention maps with the compacted RoI features. (5) Feed the selected RoI features into a low-capacity detection subnetwork for RoI classification and regression.

SPP-Net [16] and Fast R-CNN [14] firstly extract region-independent feature maps at the full-image level, and then pool region-wise features via spatial extents of proposals.

Region-Wise Feature Aggregation Networks. To improve detection accuracy, several methods try to aggregate more effective region-wise features. MR-CNN [12] develops multiple region adaption modules to pool features from a candidate box’s multi-regions. HyperNet [18] integrates hierarchical feature maps together to generate hyper RoI features. Similarly, MS-CNN [3] employs multi-scale layers for accurate proposal generation and classification. SDP [39] performs cascaded RoI pooling from different layers followed by corresponding RoI classifiers. ION [2] exploits information both inside and outside RoIs along with four-directional IRNN represented contextual information. GBD-Net [40] addresses a gated bi-directional CNN to leverage features from multiple support regions.

Region-Based Fully Convolutional Networks. After that, R-FCN [5] and Deformable R-FCN [6] encode sub-region information in the detection framework by constructing a set of position-sensitive score maps. Instead of adopting a detection subnetwork, R-FCN computes the classifica-

tion scores of sub-regions with a position-sensitive pooling layer and then averages them into the final RoI scores. R-FCN sufficiently performs multiple regression strategies for each sub-region, but fails to combine information of different sub-regions together. Figure 2 shows that, R-FCN well maintains partial characteristics but has a hard time keeping global structural information of objects.

Inspired by R-FCN, our feature selective network exploits sub-region information by generating sub-region attention maps for selecting RoI features. Along with aspect ratio attention maps, these translation-variant attention maps empower RoI features to form operative feature representations for objects.

3. Feature Selective Network

3.1. General Architecture

Our goal is to extract effective RoI features from **translation-invariant** convolutional feature maps. To achieve this goal, we adopt the popular two-stage object detection framework that consists of a proposal generator and a region classifier. Figure 3 shows an overview of

our network architecture. Our network firstly forwards an input image through ConvNet and produces the convolutional feature map of the entire image. After that, the Region Proposal Network (RPN) [31] is grafted for generating candidate RoIs by anchor box classification and regression. Given the feature map and RoIs, we focus on the design of RoI feature extractor.

Our RoI feature extractor consists of RoI feature re-extraction and region-wise attention generation. Before region-wise attention generation, we adopt a 1×1 convolutional layer to reduce the channel number to C_s and pool the compacted RoI features. Each RoI is assumed to be divided into N_{sr} sub-regions for customized sub-region attention map extraction. According to sub-region division ($N_{sr} = 3 \times 3$ demonstrated in Figure 3), we generate an $N_{sr}C_s$ -d sub-region attention bank for the entire image by a group of designed shifted convolutional layers. Likewise, we classify RoIs of different aspect ratios into N_{ar} categories ($N_{ar} = 3$ demonstrated in Figure 3) and then generate an $N_{ar}C_s$ -d aspect ratio attention bank.

Once provided the detailed information of an RoI, two designed selective RoI pooling layers max-pool the active attention maps from particular channel ranges in the sub-region attention bank and the aspect ratio attention bank, respectively. We aggregate the sub-region attention map M_{sr} and aspect ratio attention map M_{ar} together into the translation-variant attention map by element-wise addition. Finally, we merge the attention map with the compacted RoI features by element-wise product to get the selected RoI features.

$$\hat{f}_i = f_i \cdot (M_{sr_i} + M_{ar_i})$$

where $i = 1, \dots, N$, and N is the number of RoIs. For i -th RoI, f_i is the compacted RoI features, and \hat{f}_i is the selected RoI features. M_{sr_i} and M_{ar_i} correspond to its sub-region and aspect ratio attention maps.

A subsequent low-capacity detection subnetwork outputs the RoI classification score and class-agnostic box regression offsets. With efficient translation-variant RoI features, our network achieves state-of-the-art detection performance while maintaining a small parameter number and fast inference speed.

3.2. Dimension Reduction

Traditional object detectors employ ImageNet pre-trained classification backbones to extract region-independent features, followed by region-wise MLPs for RoI classification. The classical RoI pooling layer that acts as an RoI feature extractor generates fixed-length RoI features, whose channel number is huge without dimension reduction. Compared to the prevalent object detectors, our network re-extracts RoI features with a noticeably smaller channel number. We adopt a 1×1 convolutional layer to

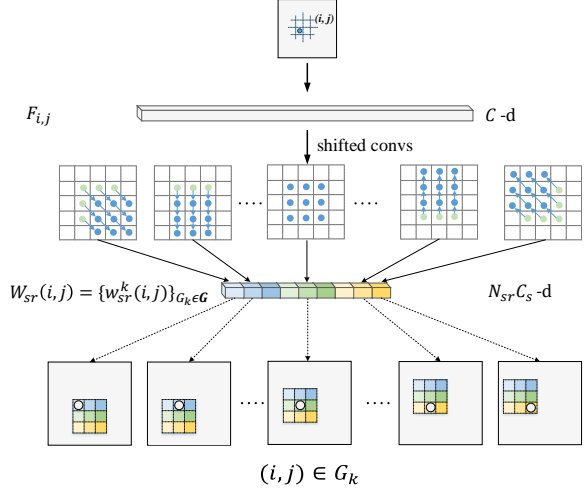


Figure 4. Illustration of sub-region attention bank's generation.

reduce the channel number of feature map from C to C_s and max-pool the RoI portion spatially into the original compacted RoI features. These compacted RoI features are expected to be optimized by later translation-variant attention maps.

3.3. Attention Banks

In order to develop translation-variant feature selection, we predict attention banks, which store all possible attention maps of spatial points when they are located in different sub-regions or RoIs of different aspect ratios.

3.3.1 Sub-Region Attention Bank

Objects for detection usually exhibit distinct spatial characteristics in different sub-regions of an RoI (boundary, inner texture, surrounding context, etc.). However, features of these sub-regions are extracted position-independently by RoI pooling in most existing object detection methods. To address this issue, we generate a sub-region attention bank. In the sub-region attention bank, the attention vector at each position comprises of position-dependent components associated with its possible locations in RoI sub-regions.

The sub-region attention bank's generation process is illustrated in Figure 4. An RoI denoted as G is divided into $N_{sr} = 3 \times 3$ sub-regions, represented by $\{G_k\}_{1 \leq k \leq N_{sr}}$. The feature vector at the (i, j) -th spatial position of convolutional feature map is denoted as $F_{i,j}$ with C channels. The active sub-region attention at (i, j) are obtained by,

$$w_{sr}^k(i, j) = \Phi_k(F_{i,j})$$

$w_{sr}^k(i, j)$ is the active attention vector with C_s channels assuming that the position (i, j) is located in the spatial extent

of k -th sub-region G_k of an RoI. Φ_k represents the feature attention extractor for G_k , which is implemented in the form of the k -th shifted convolutional layer. The parameters of Φ_k are learned by backpropagation. The sub-region attention bank W_{sr} at the position (i, j) is a catenation of active feature attentions $\{w_{sr}^k(i, j)\}_{1 \leq k \leq N_{sr}}$. Hence, the total size of sub-region attention bank is $H \times W \times N_{sr} C_s$. H and W are the spatial sizes of the original convolutional feature map. The total channel number of sub-region attention bank is $N_{sr} C_s$.

On the original convolutional feature map, every spatial position shares the same representations encoded in C channels. Instead, in the sub-region attention bank, every spatial position has a set of customized attention values containing sub-region details. For the spatial position (i, j) , channels ranging from $(k - 1)C_s + 1$ to kC_s indicate the specific feature attentions when (i, j) is located in the spatial extent of the k -th sub-region of an RoI.

For a concrete example, when setting the selective channel number to 40, we generate an attention bank with a total of 360 channels. Each set of 40 channels for a spatial position serves as the active attention values when the position is inside an RoI's sub-regions of *top-left*, *top-center*, *top-right*, ..., *bottom-center* and *bottom-right*, respectively. Intuitively, in the following feature-selective RoI pooling layer, each feature map point inside an RoI will correspond to a group of 40 channels in the sub-region attention bank, according to the detailed relative position.

Shifted Convolution. We design a group of shifted convolutional layers on the convolutional feature map of the entire image to produce the sub-region aware attention bank. The shifted convolutions are special cases of deformable convolutions in [6]. Different from deformable convolution, shifted convolution keeps the same 2D offsets for different spatial positions on feature map. As shown in Figure 4, the 2D offsets of shifted convolutional layers are fixed to $(1, 1)$, $(1, 0)$, $(1, -1)$, ..., $(-1, -1)$, respectively. The shift directions are aligned with the directions that the corresponding sub-regions towards the RoI center. The convolution kernel is fixed to 3×3 .

3.3.2 Aspect Ratio Attention Bank

Sub-region attention bank explores the potentials of position information inside RoIs. Apart from sub-regions, our network also takes into account aspect ratio information. In practice objects of different categories usually carry different aspect ratios. Besides objects difference, aspect ratio information may also reflect the viewpoint or pose of an object. For instance, there are large differences in aspect ratio between pedestrians and sitting people. Therefore, aspect ratio information should also be considered in RoI feature extraction. However, the classical RoI pooling layer gener-

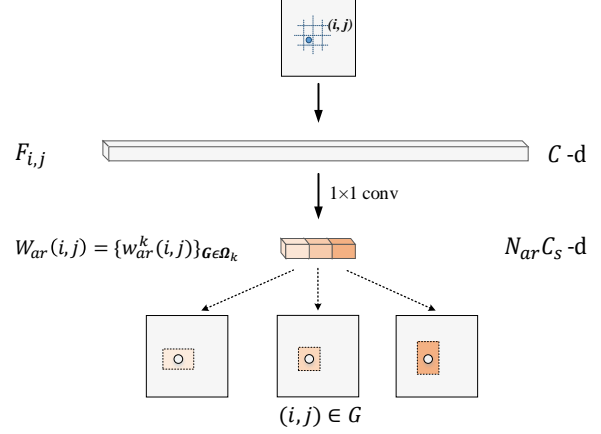


Figure 5. Illustration of aspect ratio attention bank's generation.

ates a fixed spatial size (e.g., 7×7) feature representation for all RoIs, thereby ignoring the aspect ratio difference between them.

To remedy this issue, we produce the aspect ratio attention bank to utilize the aspect ratio information. In parallel with sub-region attention bank, a 1×1 convolutional layer is placed on the convolutional feature map to get the aspect ratio aware components of each spatial position. The generation process is shown in Figure 5. We group RoIs of different aspect ratios into N_{ar} categories (N_{ar} is set to 3 in Figure 5: ratio < 0.75 , ratio > 1.3 and others). w_{ar}^k with C_s channels is derived from $F_{i,j}$, inferring the active feature attention if the position (i, j) is located in the RoI categorized to the k -th RoI set Ω_k . The aspect ratio attention bank W_{ar} at (i, j) is a catenation of active feature attentions $\{w_{ar}^k(i, j)\}_{1 \leq k \leq N_{ar}}$.

3.4. Attention Maps

In our feature selective network, attention banks are produced to reveal the particular and distinct characteristics of different sub-regions and aspect ratios. Once candidate RoIs are provided by RPN, we perform *selective RoI pooling* layers to dedicate the sub-region and aspect ratio details to translation-variant attention maps. Our selective RoI pooling layer leverages max-pooling to map an RoI's spatial extent on attention banks into a fixed-length attention vector of $h \times w \times C_s$ (e.g., $7 \times 7 \times 40$). h and w are the pooling sizes, and C_s is the selective channel number.

Selective RoI Pooling. Specifically, in an RoI window, (m, n) -th pooling bin has a correspondence with indexes k_{sr} ($1 \leq k \leq N_{sr}$) and k_{ar} ($1 \leq k \leq N_{ar}$) for sub-region attention bank and aspect ratio attention bank, respectively. For the sub-region attention bank, index k_{sr} denotes that most area of the pooling bin belong to the spatial rectangular $G_{k_{sr}}$ of the k_{sr} -th sub-region. Generally, $\text{bin}(1, 1) \in G_1, \dots, \text{bin}(m, n) \in G_{k_{sr}}, \dots, \text{bin}(h, w) \in G_{N_{sr}}$.

For the aspect ratio bank, index k_{ar} denotes that the RoI is a member of k_{ar} -th aspect ratio set $\Omega_{k_{ar}}$. Consequently, $h \times w$ pooling bins share the same k_{ar} . Given k_{sr} and k_{ar} , we max-pool the attention values from particular channel lists of the spatial positions inside each pooling bin.

$$M_{sr}(m, n, c) = \max_{(i,j) \in \text{bin}(m,n)} W_{sr}(i, j, c + (k_{sr} - 1)C_s)$$

$$M_{ar}(m, n, c) = \max_{(i,j) \in \text{bin}(m,n)} W_{ar}(i, j, c + (k_{ar} - 1)C_s)$$

where $M_{sr}(m, n, c)$ and $M_{ar}(m, n, c)$ respectively stand for the c -th channel of the (m, n) -th pooling bin for sub-region attention map and aspect ratio attention map ($1 \leq m \leq h, 1 \leq n \leq w, 1 \leq c \leq C_s$). According to the indexes k_{sr} and k_{ar} , the particular channel lists range from $1 + (k_{sr} - 1)C_s$ to $k_{sr}C_s$ in the sub-region attention bank W_{sr} and $1 + (k_{ar} - 1)C_s$ to $k_{ar}C_s$ in the aspect ratio attention bank W_{ar} , separately.

Hence, each pooling bin of the RoI selects a distinctive C_s -d sub-region dependent attention map. RoIs of different aspect ratios also select a particular C_s -d aspect ratio aware attention map. So far, we produce two attention maps with fixed-length of $h \times w \times C_s$ as the representations of sub-region and aspect ratio disparities. We merge the two maps to get the translation-variant attention map by element-wise addition. After weighting the original dimension-reduced RoI features, we feed the selected RoI features into a lightweight detection subnetwork to get the classification and regression outputs.

During backpropagation, the selective RoI pooling layer follows the chain rule and propagates the gradients from attention maps to the specific position in attention banks, according to the aforementioned correspondence.

3.5. Detection Subnetwork

In general, the detection subnetworks of two-stage object detectors are high-capacity or deep. The high-capacity ones are originated from AlexNet and VGG, which have two 4096-d fc layers designed for image classification. For detectors adopting superior fully convolutional classification backbones (e.g., ResNet, GoogLeNet), a deep and convolutional detection subnetwork is necessary [32].

Enabled by concise and informative RoI feature re-extraction, we simplify our detection subnetwork to a single low-capacity (e.g., 500-d) fc layer, followed by a $(cls+1)$ -d fc layer to get RoI classification scores and a 4-d fc layer to output class-agnostic bounding box regression offsets. Our detection subnetwork has a much smaller parameter size than the classical RoI classifier. At the same time, this simple design overcomes the drawback of costing inference time when the detection subnetwork has a deep network architecture.

4. Experiments

Our feature selective network is a generic and effective method of extracting RoI features for two-stage object detectors. In the subsequent experiments, we attach our network to commonly used ConvNet backbones: ResNet-101, GoogLeNet and VGG-16.

4.1. Experimental Setup

ConvNet Backbones. Original ResNet-101 and GoogLeNet, designed for image classification, are fully convolutional and have a stride of 32 pixels on the last convolutional layers. We follow the modification in NoC [32]: reducing the stride from 32 to 16 by changing the last stride operation from 2 to 1 and applying the “hole algorithm” [26, 4](“*Algorithme à trous*” [28]). We then attach our network to the last convolutional layers of ConvNet backbones.

Proposal Subnetwork. For region proposal generation, we use RPN to generate 300 proposals per image, and then perform all ablation experiments on fixed proposals obtained from the RPN of three ConvNets, respectively.

Detection Subnetwork. We feed the RoI feature into one added fc layer for feature reorganization, followed by a $(cls+1)$ -d fc layer and a 4-d fc layer to get outputs. The dimension of added fc layer is 500-d, or changed to 100-d if RoI features only have one channel. We use softmax loss and smooth L_1 loss defined in Fast R-CNN [14] for back-propagation [21].

Training Details. We start from ImageNet pre-trained models of each ConvNet, and label an RoI as foreground when its Intersection over Union (IoU) overlap with a ground truth box is at least 0.5. Other RoIs are labeled as background. During the training process, we construct each minibatch from two images with 256 RoIs and allocate 25% of them to the foreground. Image or its horizontal flip are resized to a single scale that its shorter side has 600 pixels.

Testing Details. We test all images on single scale with shorter side 600 pixels, and adopt non-maximum suppression (NMS) with an IoU threshold of 0.3, then evaluate the results on the test benchmarks: PASCAL VOC 2007 test, VOC 2012 test, and MS COCO val2015 sets.

4.2. Compared to baselines

ResNet-101. We implement our algorithm with ResNet-101 on three settings. We divide RoIs into 3×3 sub-regions and three aspect ratio groups (ratio < 0.75 , ratio > 1.3 and others). We generate attention banks by convolutional layers after res5c, and then select 40 channels for each pooling bin during feature-selective RoI pooling layers. To facilitate comparison, Ours(a) only uses sub-region attention map, Ours(b) only applies aspect ratio attention map, and Ours(c) combines sub-region attention map and aspect ratio

method	sub-region?	aspect ratio?	mAP(%)
Faster R-CNN			78.8
R-FCN [5]			79.5
Ours(a)	✓		82.2
Ours(b)		✓	81.0
Ours(c)	✓	✓	82.9

Table 1. Detection results on VOC 2007 test set using ResNet-101. The training set is VOC 07+12 trainval.

method	Network	sub-region?	aspect ratio?	mAP (%)
Faster R-CNN	GoogLeNet			74.8
Ours(a)	GoogLeNet	✓		76.4
Ours(b)	GoogLeNet		✓	75.6
Ours(c)	GoogLeNet	✓	✓	76.8
Faster R-CNN	VGG-16			73.2
Ours(a)	VGG-16	✓		73.9
Ours(b)	VGG-16		✓	73.6
Ours(c)	VGG-16	✓	✓	74.3

Table 2. Detection results on VOC 2007 test set using GoogLeNet and VGG-16. The training set is VOC 07+12 trainval.

attention map together. Table 1 shows the VOC 2007 test results of the reimplemented Faster R-CNN, R-FCN and our network using ResNet-101. R-FCN [5] with OHEM achieves 79.5% mAP. By contrast, Ours(a) with only sub-region information improves the baselines by 2.7 points. With the assistance of aspect ratio information, we further gain 82.9% mAP, outperforming the baselines over 3 points. These results verify the effectiveness of the region-wise attention maps generated by our feature selective networks.

GoogLeNet and VGG-16. We conduct more experiments on GoogLeNet and VGG-16 with the same settings as in ResNet-101. We attach our network to the last convolutional layers: inception5b and conv5_3, then extract $7 \times 7 \times 40$ -d features for each RoI. To reimplement Faster R-CNN with GoogLeNet, we insert an RoI pooling layer after inception4e and employ later convolutional layers as detection subnetwork. Table 2 shows the VOC 2007 test results on GoogLeNet and VGG-16. Faster R-CNN with GoogLeNet achieves 74.8% mAP. In comparison, our network achieves 76.8%, 2.0 points higher than the baseline. For the VGG-16 network, our method also yields a slight improvement. Notice that, Faster R-CNN inherits the two high-capacity 4096-d fc layers from VGG-16, thus has a much larger model size than our detector. The experiments on GoogLeNet and VGG-16 indicate that the feature selective module is generic and robust to prevalent ConvNets. The ranges of performance improvements reflect the feature representation potential of each ConvNet, and our feature selective modules exploit the potentials to some extent.

method	sub-region?	aspect ratio?	mAP(%)
Faster R-CNN			73.8
R-FCN			77.6
Ours(a)	✓		79.5
Ours(b)		✓	77.9
Ours(c)	✓	✓	80.5

Table 3. Detection results on VOC 2012 test set using ResNet-101. The training set is VOC 07+12 trainval+07 test.

method	$AP@0.5$	AP	AP_s	AP_m	AP_l
Faster R-CNN	48.4	27.2	6.6	28.6	45.0
R-FCN	48.9	27.6	8.9	30.5	42.0
Ours	54.0	33.6	17.8	35.4	46.5

Table 4. Detection results on the MS COCO val 2015 set using ResNet-101. The training set is COCO train 2015.

method	# params	inference speed (sec/img)	mAP (%)
Faster R-CNN	40.7M	0.31	78.7
R-FCN	44.0M	0.12	79.5
Ours (1x1 conv)	42.6M	0.19	82.3

Table 5. Comparisons in parameter number, inference speed and detection accuracy on VOC 2007 test set using ResNet-101. The training set is VOC 07+12 trainval.

4.3. VOC 2012 Results

Table 3 shows the results of VOC 2012 test using ResNet-101. We evaluate the performances with the same settings as in VOC 2007 test, dividing RoIs to 3×3 sub-regions, 3 aspect ratio ranges and selecting 40 channels for RoI features. R-FCN [5] with multiscale training and OHEM [35] obtains 77.6% mAP. Compared to that, our method achieves 80.5% mAP.

4.4. MS COCO Results

We perform experiments on the challenging MS COCO dataset that has 80k train images and 40k val images. Since COCO has more complicated object categories, we make a modification that selectively pool 80 channels for each RoI. For the detection subnetwork, we repurpose the dimension of the fc layer into 800-d. Table 4 shows that our network achieves a better performance (54.0% / 33.6%) compared to Faster R-CNN [17] (48.4% / 27.2%) and R-FCN [5] (48.9% / 27.6%) baselines. Our feature selective network far surpasses other methods on small size object detection.

4.5. Parameter Number and Inference Speed

Table 5 shows the overall comparisons in parameter number, inference speed and the performance. Here our network adopts 1×1 convolution as the trade-off between

feature	channel	RoI classifier	mAP(%)
res4b22	1024	res5a, 5b, 5c, fc21	78.7
res5c	2048	fc4096, fc4096, fc21	78.8
res5c	2048	fc500, fc21	78.4
res5c	40	fc500, fc21	79.5

Table 6. Detection results on VOC 2007 test set using Faster R-CNN with ResNet-101. The training set is VOC 07+12 trainval.

performance and model complexity when generating sub-region attention map. The selective channel number is set to 40. We record the inference speeds on a Titan X GPU. Our network gains a better performance while maintaining a smaller parameter number compared to R-FCN. On the other hand, the deep detection subnetwork leads to significant per-RoI computation cost for Faster R-CNN. With the simplified detection subnetwork, our network has a higher inference speed than Faster R-CNN.

5. Ablation studies

5.1. Dimension Reduction

Before region attention map generation, our feature selective network adopts a 1×1 convolutional layer to reduce the channel number of convolutional feature map and RoI-pools the original compacted RoI features. Here we perform the ablation experiments on the role of dimension reduction. We reimplement Faster R-CNN with ResNet-101 using the two settings in [32]: one uses res4b22’s output as RoI feature and employ conv5 block as detection network, the other one adds two 4096-d fc layer as RoI classifier.

Table 6 shows that, with a simplified detection subnetwork, dimension reduction achieves 79.5% mAP on VOC 2007 test set, which is a 0.7% performance improvement than the baseline. This indicates that a lightweight RoI classifier with compacted RoI features may receive a better detection result.

5.2. Selective Channel Number

Our feature selective network ensures a promising detection accuracy with a small channel number of RoI features. Here we investigate on the role of selective channel number C_s . We follow the same settings as before but change the selective channel number from 100 to 1 and report the results. Table 7 shows that, enabled by sub-region and aspect ratio attention maps, our networks have similar results when the selective channel number C_s varies from 20 to 100. 20 channels seem to be enough for RoI feature representation. Surprisingly, even reduced to one channel, the re-extracted RoI feature with a fixed-length of 49 still performs well.

channel number C_s	100	40	20	5	1
mAP(%)	82.9	82.9	82.5	80.5	79.4

Table 7. Detection results on VOC 2007 test set using ResNet-101 with different selective channel numbers. The training set is VOC 07+12 trainval.

conv kernel	shifted ?	shift direction	mAP(%)
1×1		-	82.3
3×3		-	82.6
3×3	✓	center	82.9
3×3	✓	outside	82.5
3×3	✓	random	82.5

Table 8. Detection results on VOC 2007 test set using ResNet-101 with different convolution settings. The training set is VOC 07+12 trainval.

5.3. Shifted Convolution

Our feature selective network adopts shifted convolution operation when generating sub-region attention bank. The shifted convolution enables different feature extraction ways when a feature map point is inside different sub-regions. Here we investigate on the influence of shifted convolution operation. We adopt 1×1 and 3×3 standard convolution when generating sub-region attention bank. The results in table 8 shows that, 3×3 convolution yields 0.3% mAP gain than 1×1 benefitting from the increase of parameters. When equipped with shifted convolution, our network further achieves a 0.3% mAP gain. The shift direction is towards the assumed RoI’s center. If we change the shift direction to the opposite or random, evolving more features outside the RoI, the shifted convolution could hardly achieve better results than standard convolution. These results indicate that when extracting sub-region feature attentions for RoI, the feature information inside an object may play a more important role.

6. Conclusion

We propose feature selective networks to distill effective RoI features from convolutional feature maps. By generating attention banks, we exploit the translation-variant potential of RoI feature representations. Based on the detailed sub-region and aspect ratio information, distinctive attention maps are selected for each RoI and used to refine the original compacted RoI features. With a surprisingly small channel number 1-d for RoI features, our feature selective network ensures a state-of-the-art detection accuracy. With a proper selective channel number, our networks further achieve general improvements equipped with the prevalent ConvNet backbones (ResNet-101, GoogLeNet and VGG-16). Our method offers a general and efficient module to dedicate RoI preference to object detection networks.

References

- [1] H. Azizpour and I. Laptev. Object detection using strongly-supervised deformable part models. In *ECCV*, 2012. 2
- [2] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *CVPR*, 2016. 3
- [3] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV*, 2016. 3
- [4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 6
- [5] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. 2, 3, 7
- [6] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *ICCV*, 2017. 3, 5
- [7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1
- [8] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. In *TPAMI*, 2014. 2
- [9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 1, 2
- [10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. In *TPAMI*, 2010. 2
- [11] A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, and L. Van Gool. Deepproposal: Hunting objects by cascading deep convolutional layers. In *ICCV*, 2015. 2
- [12] S. Gidaris and N. Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *ICCV*, 2015. 2, 3
- [13] S. Gidaris and N. Komodakis. Locnet: Improving localization accuracy for object detection. In *CVPR*, 2016. 2
- [14] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 1, 3, 6
- [15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 3
- [17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 7
- [18] T. Kong, A. Yao, Y. Chen, and F. Sun. Hypernet: towards accurate region proposal generation and joint object detection. In *CVPR*, 2016. 1, 3
- [19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1
- [20] W. Kuo, B. Hariharan, and J. Malik. Deepbox: Learning objectness with convolutional networks. In *ICCV*, 2015. 2
- [21] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989. 6
- [22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1
- [23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1
- [24] 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. In *ECCV*, 2014. 1, 2
- [25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 1
- [26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 6
- [27] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1
- [28] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999. 6
- [29] A. M. Øygard. Deepdraw. DeepDraw on <https://github.com/auduno/deepdraw>, 2015. 2
- [30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1
- [31] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 4
- [32] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. In *TPAMI*, 2016. 6, 8
- [33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 1, 2
- [34] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014. 2
- [35] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016. 7
- [36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1, 2
- [37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 1, 2
- [38] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 2013. 2
- [39] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *CVPR*, 2016. 3
- [40] X. Zeng, W. Ouyang, B. Yang, J. Yan, and X. Wang. Gated bi-directional cnn for object detection. In *ECCV*, pages 354–369, 2016. 2, 3
- [41] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 2