Machine Vision and Applications (2019) 30:543–555
https://doi.org/10.1007/s00138-019-01017-9

SPECIAL ISSUE PAPER

Gated bidirectional feature pyramid network for accurate one-shot detection

Sanghyun Woo[1] · Soonmin Hwang[1] · Ho-Deok Jang[1] · In So Kweon[1]

Received: 31 July 2018 / Revised: 18 December 2018 / Accepted: 17 February 2019 / Published online: 13 March 2019

Abstract
Despite recent advances in machine learning, it is still challenging to realize real-time and accurate detection in images. The recently proposed StairNet detector (Sanghyun et al. in Proceedings of winter conference on applications of computer vision (WACV), 2018), one of the strongest one-stage detectors, tackles this issue by using a SSD in conjunction with a top-down enrichment module. However, the StairNet approach misses the finer localization information which can be obtained from the lower layer and lacks a feature selection mechanism, which can lead to suboptimal features during the merging step. In this paper, we propose what is termed the gated bidirectional feature pyramid network (GBFPN), a simple and effective architecture that provides a significant improvement over the baseline model, StairNet. The overall network is composed of three parts: a *bottom-up pathway*, a *top-down pathway*, and a *gating module*. Given the multi-scale feature pyramid of deep convolutional network, two separate pathways introduce both finer localization cues and high-level semantics. In each pathway, the gating module dynamically re-weights the features before the combining step, transmitting only the informative features. Placing GBFPN on top of a basic one-stage detector SSD, our method shows state-of-the-art results.

Keywords Object detection · Multi-scale aggregation · One-shot detection

1 Introduction

Recognizing objects accurately while maintaining a high speed is a fundamental challenge in computer vision. One of the earliest approaches [1] computes the feature pyramid from multi-scale images to enable a detector to search different spatial locations as well as pyramid levels. Classic object detectors such as DPM [13] based on hand-engineered features [10,35] rely heavily on this standard method to produce fine results. However, this approach requires a large memory footprint, and using it increases the inference time significantly, both of which are undesirable for actual applications. After the introduction of deep convolutional neural networks (CNNs), engineered features were replaced with richer CNN features and detection systems [9,41,43] changed and now use only single-scale features for more rapid detection. Although CNN features are capable of representing strong semantics and are also more robust with regard to scale variations, such techniques are suboptimal given the overreliance on single features.

Recently, the use of the inherent multi-scale features of deep CNNs has been proposed as an effective alternative. Liu et al. [33] suggest SSD, which uses different layers of CNN features that are computed during the forward pass of a single resolution input image (Fig. 1a). These features have different spatial resolutions and form a feature pyramid which approximates another method [1] in a cost-free manner. However, given that the CNN encodes different levels of semantics across the depth, directly using the pure internal pyramid is a suboptimal solution as well. Thus, utilizing extra convolution blocks before the prediction layer, which deepens the network and improves feature representation, has been suggested, as doing so avoids the direct use of a pure feature pyramid (Fig. 1b) [20,56]. However, this approach misses the

✉ Sanghyun Woo
shwoo93@kaist.ac.kr

Soonmin Hwang
jjang9hsm@kaist.ac.kr

Ho-Deok Jang
hdjang@kaist.ac.kr

In So Kweon
iskweon77@kaist.ac.kr

[1] Department of Electrical Engineering, Korea Advanced Institute of Science and Technology(KAIST), Room #212, N1 Building, 291 Daehak-ro, Yuseong, Daejeon 34141, Korea

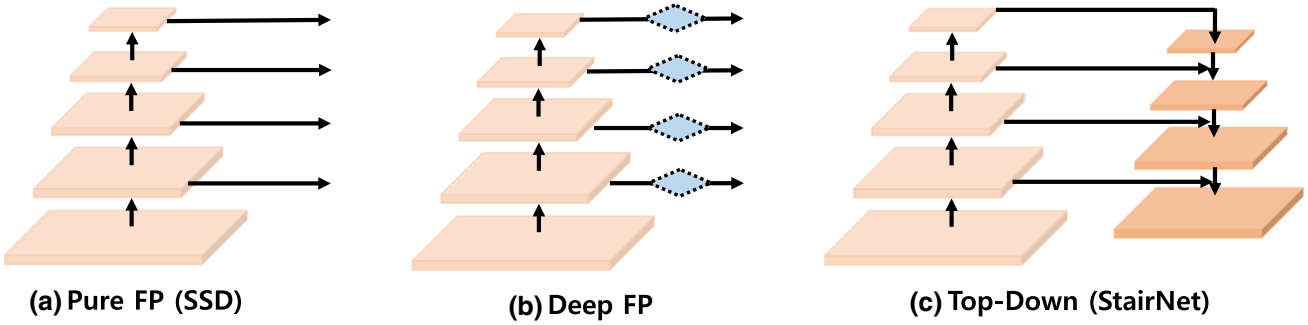

Fig. 1 Various multi-scale aggregation methods for detectors. **a** Pure feature pyramid that is computed by a forward pass of CNN is directly used for multi-scale prediction. FP denotes feature pyramid. **b** Deep FP exploits convolution blocks before the prediction layers to increase the overall depth and enhance feature representations. **c** StairNet [46] augments pure feature pyramid with a top-down pathway and enriches the semantic level of high-resolution features

opportunity to exploit meaningful signals that are distributed throughout the network. In order to resolve this issue, recent studies [14,29,31,46,49] adopt an hourglass structure (i.e., a top-down pathway) [37,45]. Intuitively, this enables a model to maintain semantically strong features using a combination of upsampling and a lateral connection. Sanghyun et al. [46] proposed StairNet, as noted above, which makes the entire system a top-down structure (hourglass). They empirically verified that the method enhances the internal feature pyramid significantly (Fig. 1c). Top-down aggregation is now a commonly adopted design choice to improve scale invariance in both two-stage [19] and one-stage detectors [11,14,32,46].

Nevertheless, StairNet has several limitations. First, the finer spatial cues which can be obtained from low levels of the network are somewhat disregarded. Given multi-scale features, StairNet adopts a unilateral information flow, from top to bottom. The top-down pathway then gradually aggregates the semantics, focusing only on adjusting the semantic levels of all features. However, because low-resolution features do not convey the accurate positions of objects due to the quantization effect of pooling, a bottom-up pathway is needed to offset the lost information (see Fig. 3). Second, StairNet lacks a gating mechanism that can adaptively transmit only meaningful information in the merging step (see Fig. 3). We empirically verify that this gating mechanism enhances the performance of the detector.

In this paper, we propose the gated bidirectional feature pyramid network (GBFPN), which fills in the missing pieces of StairNet. The network is split into three parts: the bottom-up pathway, the top-down pathway, and the gating module. Relying on this network, we construct an enhanced feature pyramid with finer localization information, with strong semantics at all levels, and which is built quickly from a single-scale image. We evaluate our method by replacing the pure feature pyramid of SSD with GBFPN. Without adopting a deep backbone networks [18,51,57] or using extra labels for training [11], we report a state-of-the-art result on the challenging benchmark datasets [12,30]. We also show that our method improves the performance significantly even with the MobileNet backbone, suggesting its great potential for applications on low-end devices. Our method can be trained end-to-end manner with all feature scales.

Contribution Our main contributions are as follows.

- We propose a novel multi-scale aggregation method (GBFPN) that can be widely applied to improve the accuracy levels of generic object detection tasks.
- We validate the effectiveness of GBFPN through extensive ablation studies.
- We verify that the performance of GBFPN-SSD is significantly improved from the baseline SSD on benchmark datasets (VOC 2007 and MS COCO) without all of the bells and whistles.

2 Related work

2.1 Convolutional object detectors

Much progress has been made in the area of object detection due to the use of CNNs. Recent work on object detectors based on these networks can be divided into two streams. The first focuses on two-stage detectors, pioneered by R-CNN [15]. These detectors rely on sparse sets of candidate regions that should contain objects proposed during the first stage [52] followed by a second stage for refinement. R-CNN detector has been improved over the years, in terms of both speed and accuracy. The faster R-CNN detector [43] integrates the proposal generation stage with the second-stage classifier into a single CNN. Thus far, these two-stage methods [9,19,31,49] have continuously filled the top entries of benchmark datasets [12,30]. Despite the fact

that they promise high accuracy, recent works [24] have shown that two-stage detectors are plagued with slow inference speeds and high-memory usage rates, indicating that they are unlikely to be suitable for both real-time applications and low-end devices. This motivates the design of rapid and lightweight one-stage detector in which a model predicts objects in a proposal-free manner.

OverFeat [47] is a first one-stage object detector that applies a CNN as a sliding window detector on an image pyramid. More recently, YOLO [41] and SSD [33] were proposed for the real-time processing. These approaches divide an image into multiple grids and predict class confidence levels and bounding boxes simultaneously. Unlike YOLO, SSD adopts in-network multi-scale features, making it more robust with varying shapes and sizes of objects. Based on this framework, numerous extensions have been proposed [11,26,42,48,64]. In this paper, we propose GBFPN, which is a simple and effective network that enhances the performance of SSD significantly.

2.2 Using multi-scale features

It is well known that utilizing multiple layers from different depths of a CNN can improve recognition tasks. SSD [33] distributes variously sized default boxes to multiple layers, enforcing each layer to predict the specific scales of objects. MS-CNN [3] adopts a similar architecture, using multiple features with deconvolution layers for detection. RRC [42] utilizes features from adjacent layers recursively. SPPNet [17], GBDNet [62], and DeepLAB [6] pool multiple scales of features using only a single-scale feature map. HyperNet [27] and ION [2] concatenate features from different layers and obtain object proposals from them. FCN [34] and Hypercolumns [16] merge partial scores for each class over multiple features for accurate segmentation. U-Net [45], Stacked Hourglass networks [37], and StairNet [31,46,63] exploit an encoder–decoder structure (i.e., hourglass) using skip connections and upsampling for segmentation, human pose estimation, and detection, respectively. The concurrent study [7] presents adaptive multi-scale information flow (ASIF) module which aggregates information repeatedly from neighboring layers in the feature pyramid. The proposed method GBFPN similarly exploits multi-scale features but is in much simpler form with additional gating mechanism and shows better performance.

2.3 Gating mechanism: attention

An interesting property of the human vision system is that it does not process an entire scene at once. Instead, humans selectively focus on salient parts in order to capture the visual structure better, using an ability called *attention* [8,25,28,44]. Based on this concept, early works attempted to embed this type of processing using RNN [36] or LSTM [21] and demonstrate its effectiveness in image/video captioning [5,59,61] and visual question answering [4,58,60] tasks. Recently, the gating mechanism has been applied to improve classification tasks [23,39,53,55]. Wang et al. [53] suggested what is termed a *residual attention network*, which utilizes a mask branch composed of hourglass attention module. The final architecture is more accurate and robust with regard to noisy inputs due to its feature refinement process. Hu et al. [23] developed a compact gating module that computes channel-wise attention, referring to it as *Squeeze and Excitation*. They show that the accuracy is consistently improved in various CNN models simply, by dropping this simple module into the network. Motivated by these previous works, we adopt a gating mechanism for use in the feature merging step (Fig. 2 right). The gating module adaptively controls the signals from the lower/higher layer of the features and selects informative features before they are merged. We empirically verify that this gating mechanism improves the performance of the detection system with only negligible overhead.

3 Method

We briefly present an overview of our method before describing in more detail. Two separate pathways, the top-down and bottom-up pathways, transmit strong semantics from layer at higher level and accurate shape and appearance cues from lower levels. In this paper, we adopt the SSD framework (Fig. 1a) as a meta-architecture to build the gated bidirectional feature pyramid network (GBFPN) on it. The proposed GBFPN is composed of three parts, as shown in Fig. 2. First, the top-down pathway takes high-level features and spreads out the strong semantics gradually until the bottom of the feature pyramid is reached. Then, to incorporate detailed localization signals, the bottom-up pathway takes low-level features and distributes their output until the top of the feature pyramid is reached. In each pathway, we position a gating module that adaptively selects meaningful features and suppresses less useful ones, controlling how the message passes before the merging step at every scales. Finally, we use element-wise summation to combine two enhanced feature pyramids for the prediction. The entire process occurs in a one-shot manner.

3.1 Bidirectional feature pyramid

Given a feature pyramid (Fig. 2 blue feature pyramid), the top-down pathway upsamples spatially coarser but semantically stronger features using a deconvolution layer. The upsampled features are then combined with the corresponding bottom features, which undergo a 1×1 convolution layer (i.e., lateral connection) by element-wise summation. This is

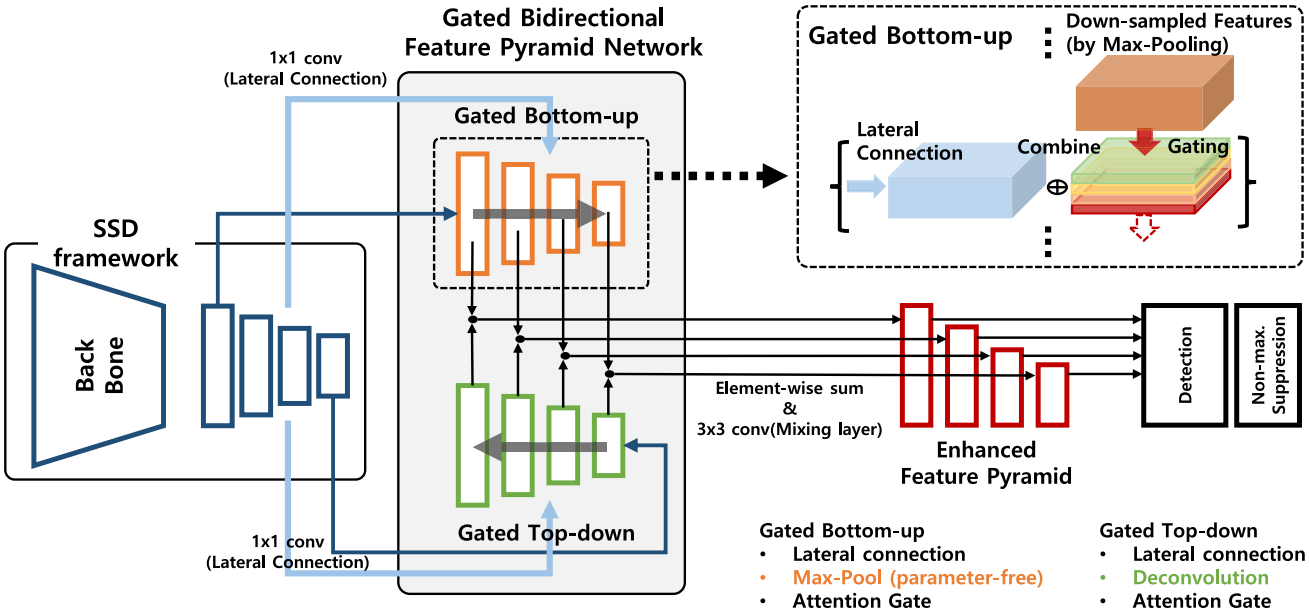

Fig. 2 The overall architecture of GBFPN. Our proposed GBFPN is composed of three parts: top-down pathway, bottom-up pathway, and gating module. First, the top-down pathway takes high-level features and distributes the strong semantics gradually until the bottom of the feature pyramid is reached. The bottom-up pathway then takes low-level features and spreads out their output until the top of the feature pyramid is reached. In each pathway, we apply a gating module that adaptively selects meaningful features and suppresses less useful ones. Finally, we use element-wise summation to combine two enhanced feature pyramids. The entire process occurs in a one-shot manner. Best viewed in color

repeated until the finest resolution features are generated. The bottom-up pathway consists of a max-pooling layer followed by a relu operation to convey accurate localization information of the target object in a parameter free manner. Symmetrical with the top-down pathway, downsampled features are then merged with the corresponding upper features, undergoing a 1×1 convolution layer by element-wise summation. This is repeated until the coarsest resolution features are produced. We fix the channel dimension in all of the feature maps. We set it to 256 in this paper; thus, all added layers have 256 channel outputs. In the experimental section (see Sect. 4), we investigate different numbers of channels and empirically confirm that the proposed model is robust in how it reduces the number of model parameters. The final outputs of both pathways are illustrated in Fig. 2, orange and green feature pyramids. These two pyramids are then combined using element-wise summation to produce final feature pyramid (see Algorithm 1).

Algorithm 1 Gated Bidirectional Feature Pyramid Network

INPUT: $\mathbf{F} = [F_0, F_1, F_2, \ldots, F_{N-1}]$
OUTPUT: $\mathbf{O} = [O_0, O_1, O_2, \ldots, O_{N-1}]$
/*Feature projection*/;
for t:=0 to N-1 do **do**
  $F_{0t} \leftarrow P_{0t}(F_t)$ /* P: 1×1 conv */
  $F_{1t} \leftarrow P_{1t}(F_t)$
end for
/*Top-down pathway*/;
$T_{N-1} \leftarrow F_{0N-1}$
for t:=N-2 to 0 do **do**
  $F_{0t} \leftarrow F_{0t} + G_{0t}(\uparrow F_{0t+1})$ /* Up-arrow: up-sampling, G: gating */
  $T_t \leftarrow F_{0t}$
end for
/*Bottom-up pathway*/;
$B_0 \leftarrow F_{10}$
for t:=1 to N-1 do **do**
  $F_{1t} \leftarrow F_{1t} + G_{1t}(\downarrow F_{1t-1})$ / *Down-arrow: down-sampling, G: gating */
  $B_t \leftarrow F_{1t}$
end for
/*Combining two feature pyramid*/;
for t:=0 to N-1 do **do**
  $O_t \leftarrow M_t(T_t + B_t)$ /* M: 3×3 conv */
end for

3.2 Gating module

While two pathways transmit useful information from both high and low levels of the network, a mechanism that dynamically denoises the features is absent. We achieve this by adopting a gating mechanism [23], as originally used for image classification tasks. We also experimented with gating modules that are different in design (e.g., different pooling methods, more layers) and observed no significant improvement. Thus, we retain the use of the original design

for simplicity. The components of the gating module are described below.

As each channel of features encodes a specific semantics, we exploit the inter-channel relationship. To aggregate the feature map in each channel, we utilize global average pooling on the feature map $\mathbf{F}$ and produce channel vector $\mathbf{F_c} \in \mathbb{R}^{C\times1\times1}$. The vector after the pooling then softly encodes global statistics in each channel. To estimate attention across channels from the channel vector $\mathbf{F_c}$, we use two fully connected layers. To reduce the parameter overhead, the hidden activation size is set to $\mathbb{R}^{C/r\times1\times1}$, where r is the reduction ratio. We set r to 16 in this paper for simplicity. In short, the channel-wise attention is computed as follows:

$$\mathbf{G}(\mathbf{F}) = \sigma(\mathbf{W_1}(\mathbf{W_0}\text{AvgPool}(\mathbf{F}) + \mathbf{b_0}) + \mathbf{b_1}), \quad (1)$$

where σ is a sigmoid function, $\mathbf{W_0} \in \mathbb{R}^{C/r\times C}$, $\mathbf{b_0} \in \mathbb{R}^{C/r}$, $\mathbf{W_1} \in \mathbb{R}^{C\times C/r}$, and $\mathbf{b_1} \in \mathbb{R}^{C}$. The computed attention vector is then multiplied by the input again for re-weighting, selecting only the informative features. This mechanism allows control over how messages pass in the two pathways, allowing the building of more discriminative features.

3.3 Gated bidirectional feature pyramid network

Our final model GBFPN is a generic solution with which to build enhanced feature pyramids inside deep CNNs. It is also very simple and easy to apply to existing detection systems. Below, we adopt our method in SSD framework in order to demonstrate the effectiveness and advantages of our method. First, we apply a bidirectional feature pyramid to replace the pure feature pyramid of SSD. We add gates in each pathway to obtain GBFPN-SSD. Other principles in SSD are inherited.

Algorithm 1 depicts the overall procedure. **P** denotes 1×1 convolution for lateral connections. The **up-arrow** and **down-arrow** denote the upsampling operation and the downsampling operation, respectively. For the upsampling operation, we adopt a deconvolution layer followed by relu, while for the downsampling operation, we use max-pooling followed by relu. **G** and **M** denote the gating operation and the 3×3 convolution, respectively.

3.4 The effectiveness of GBFPN

In Fig. 3, we visualize feature maps in GBFPN so that regards can gain a clearer understanding of the concept. We select a model that is trained on PASCAL VOC 2007 train/val and PASCAL VOC 2012 train/val with an input image size of 512. We then forward the images from the PASCAL VOC 2007 test for the analysis. We select images for each pathway and gather the feature maps from corresponding features (i.e., output of 1×1 lateral convolution) and transmitted features (i.e., upsampled or downsampled features) at a certain level of the feature pyramid. For visualization, we average the 3D feature map along the channel axis and normalized it to the 0–1 scale. The feature maps in each of the red dotted boxes and the blue dotted boxes in Fig. 3 are features belonging to the top-down pathway and the bottom-up pathway, respectively

In Fig. 3 at the top, we find that the top-down pathway transmits semantic a semantic signal to complement the features of the lower layer (i.e., see the red dotted box, where faint white features are activated by upsampled features). On the other hand, the bottom-up pathway conveys the locations of missed objects (i.e., see the blue dotted box, row1: the truncated bicycle in the bottom-left of the image, row2: the bus in the middle of the image, row3: the train in the middle of the image, row4: the sheep in the bottom of the image). The visualization of each pathway demonstrates that not only the top-down pathway but also the bottom-up pathway play important roles in constructing an enhanced feature pyramid.

In Fig. 3 at the bottom, we forward the same image to both pathways to determine how they jointly produce the final features. We observe phenomenon similar to those discussed above. For the top-down pathway, we found that it spreads out the strong semantics of salient objects to the bottom layer. Meanwhile, the bottom-up pathway transmits rough localization information of missed objects(i.e., the small-sized overlapped person in the middle of the image). This finding shows that two pathways can complement each other to produce an enhanced feature pyramid, leading to accurate object detection.

Given that the gating module forces each pathway to transmit only meaningful signals, we note that the features visualized in Fig. 3 implicitly reflect this mechanism. See row3 of the red dotted box in Fig. 3. Only objects which are not found well are complemented by the upsampled features.

4 Experiments

We evaluate the GBFPN-SSD approach on two widely used datasets: PASCAL VOC 2007 and MS COCO [12,30]. For a better apples-to-apples comparison, we create our benchmark platform based on PyTorch and reproduce the original accuracy of SSD300. Our unified framework allows for a fair comparison, as shown in Fig. 1b, c, of the methods with our network (GBFPN) while keeping all other settings the same. We perform extensive ablation studies to investigate the effectiveness of each component of GBFPN thoroughly. The metric for evaluating the detection performance is the mean average precision (mAP).

4.1 PASCAL VOC 2007

We perform ablation experiments on the PASCAL VOC 2007 test sets for a detailed analysis of the proposed GBFPN. We

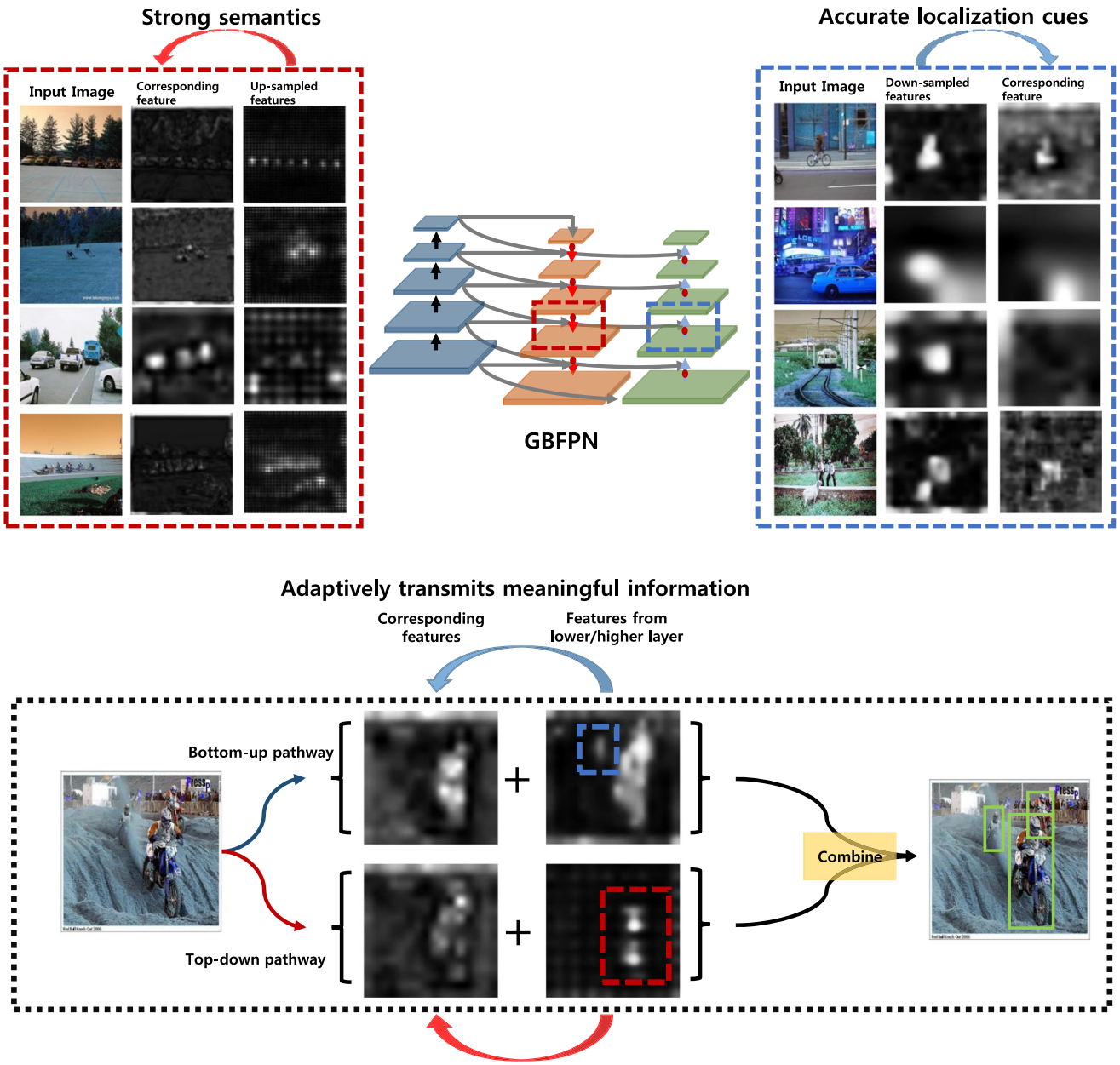

Fig. 3 The effectiveness of GBFPN: The high-level semantics and accurate localization cues are transmitted by the top-down pathway (red dotted box) and the bottom-up pathway (blue dotted box), respectively. The top-down path complements the lower layers by conveying the strong semantics of the target objects (i.e., faint white features are activated by upsampled features). On the other hand, the bottom-up path transmits missed object location cues (i.e., row1: the truncated bicycle in the bottom-left of the image, row2: the bus in the middle of the image, row3: the train in the middle of the image, row4: the sheep in the bottom of the image). Meanwhile, the gating modules force both pathways to transmit only informative signals. We note that the features visualized above implicitly reflect this mechanism (e.g., row3 of the red dotted box, where only objects which are not found well are complemented by the upsampled features). These three components of GBFPN interact to output the final enhanced feature pyramid. The outputs of both pathways are then combined by means of element-wise summation. We empirically confirm that our method outperforms previous approaches [31,33] which utilize the feature pyramid of deep CNNs for object detection. Best viewed in color (colour figure online)

train the models on the united set of VOC 2007 train/val and VOC 2012 train/val (07+12) and evaluate it on the VOC 2007 test set. We set the batch size to 32. For stable training, we use a warm-up strategy that gradually increases the learning rate from 10^{-6} to 4×10^{-3} at the first five epochs. Subsequently, it goes back to the original learning rate schedule divided by 10 at 150 and 200 epochs. The total number of training epochs is 250. We use a weight decay of 0.0005 and a momentum value of 0.9.

Table 1 Comparison with previous approaches

Method	Params	mAP	fps
SSD300* [33]	26.5M	77.8	125**
+ResBlock [18]	28.4M	78.6	100**
+InceptionBlock [50]	33.8M	78.8	66**
+ResNextBlock [57]	28.4M	78.5	83**
+SE module [23]	26.7M	77.6	125**
+NL module [54]	29.9M	78.2	90**
+Bottom-up	30M	78.9	111**
+Top-down(StairNet) [46]	33M	79.0	90**
+BFPN	33.6M	79.6	90**
+GBFPN	33.7M	79.9	83**

In contrast to previous methods [i.e., the Deep FP method (Fig. 1b) and the top-down method (Fig. 1c)], utilizing both the top-down and the bottom-up pathways is effective (BFPN). In addition, dynamic feature refinement mechanism further boosts the performance (GBFPN). *Tested in voc2007 test*

4.1.1 Comparison with previous approaches

In this experiment, we empirically verify that a significant performance improvement from the pure feature pyramid (Fig. 1a) can be achieved using two pathways and the gating module. We compare our network GBFPN with previous approaches (i.e., Fig. 1b, c). To determine the performance improvement when using the method in Fig. 1b, we add auxiliary convolution blocks [18,23,50,54,57] which have different characteristics before the prediction layer. For the method in Fig. 1c, we follow an architecture design identical to that in an earlier study [31] except for the upsampling layer, adopting a deconvolution layer instead. As shown in Table 1, we can clearly note that exploiting information from the other layers (i.e., either the bottom-up pathway or the top-down pathway) produces better performance than placing extra convolution blocks, demonstrating that inter-feature information transfers are more important than deepening the relevant pathway for accurate predictions. We also observe that the bottom-up pathway, which is composed of 1×1 convolution layers, max-pooling, and the relu operation, generates a similar performance enhancement similar to that of the top-down pathway. For a thorough analysis of the accuracy improvement of both pathways, we visualize the feature maps. As shown in Fig. 3, the bottom-up pathway transmits meaningful cues such as areas missed from the top layer due to the sub-sampling process, while the top-down pathway transmits strong semantics, ensuring that the final features to focus on the target objects more confidently.

In contrast to the previous methods, we use two complementary pathways jointly and push the performance further, demonstrating the effectiveness of using both pathways simultaneously. Moreover, adding the gating module which controls how the message passes in each pathway also further boosts the performance, achieving a mAP of 79.9%. Note that while the three different methods of SSD300 + InceptionBlock (Fig. 1b), SSD300 + the Top-down pathway (Fig. 2c), and SSD300 + GBFPN have similar numbers of parameters, GBFPN outperforms all of these methods with comparable inference speed.

Table 2 Upsampling and downsampling methods

Exp	Top method		Bot method					
	bi	deconv	conv	avg	max	Params(M)	mAP	fps
1	✓		✓			33.7	79.6	83
2	✓			✓		30.8	79.8	91
3	✓				✓	30.8	79.5	91
4		✓	✓			36.7	79.9	71
5		✓		✓		33.7	79.6	83
Ours		✓			✓	33.7	79.9	83

4.1.2 Upsampling and downsampling methods

In this experiment, we utilize different methods of upsampling and downsampling. For the upsampling, there are two ways to enhance the resolution of the feature map. The first involves the deconvolution layer in which the upsampling weights are learned through the training process. Second, naive upsampling methods such as the nearest neighbor or bilinear interpolation can be used. As shown in Table 2, a deconvolution layer in general leads to better results. This implies that the learned upsampling weights perform better than the naive upsampling kernels. Moreover, recent studies [14,38] have shown that the sequence of deconvolution layer is suitable for propagating information efficiently. For downsampling, there are three ways to reduce the resolution of the feature map. First is to use a 3×3 convolution layer in which the weights are learned, similar to a deconvolution layer. Second is to use avg-pooling and third, max-pooling. We observe that max pooling shows the best results while imposing less overhead across the network. Based on the result in Table 1, we use deconvolution as the upsampling method and max-pooling as the downsampling method in the GBFPN.

4.1.3 Comparison with ensemble baselines

We can naturally consider that forward propagation of ConvNet itself is already a type of bottom-up reasoning and that the improvement of combining two pathways is due to an ensemble effect. To address this concern, we conduct additional experiments to compare our method with ensemble baselines. Table 3 summarizes the results, showing that the

Table 3 Comparison with ensemble baselines

Method	Params	mAP	fps
SSD300	26.5M	77.8	125
+ *T* (deconv)	33M	79.0	91
+ *B* (max)	30M	78.9	111
+ Two *T* (deconv)	30.7M	79.2	77
+ Two *B* (max)	36.6M	79.1	91
+ GBFPN (deconv/max)	33.7M	79.9	83
+ *T* (bilinear)	30M	78.9	111
+ *B* (avg)	30M	79.1	111
+ Two *T* (bilinear)	30.7M	78.8	83
+ Two *B* (avg)	30.7M	79.3	91
+ GBFPN (bilinear/avg)	30.8M	79.8	91

T and B denote top-down pathway and bottom-up pathway, respectively. Parentheses include upsampling and downsampling methods accordingly. *Tested in voc2007 test*

Table 4 Combining methods of two feature pyramids

Combining method	Params	mAP
Concat-conv	37.3M	79.9
MAX	33.7M	79.4
PROD	33.7M	79.8
SUM	33.7M	79.9
Base (bottom-up)	30M	78.9
Base (top-down)	33M	79.0

performance improvement of GBFPN does not come from a mere ensemble effect but from complementary action. For the bilinear upsampling case, two top-down pathways even induce optimization difficulties. This indicates that constructing an explicit shortcut that propagates low-level signals complements the missing localization cues of the top-down pathway, building finer multi-scale features. Moreover, we observe no overfitting as well when adding the bottom-up pathway (see Table 3).

4.1.4 Combining methods of two feature pyramids

We investigate four different combining strategies in order to merge two feature pyramids (Fig. 3): concatenation followed by convolution, an element-wise maximum, an element-wise product, and element-wise summation. Table 4 summarizes the results of the comparison of the four different implementations. We confirm that element-wise summation is most effective method considering both the parameter overheads and accuracy. In terms of the information flow, the element-wise summation is an effective means of securing and integrating the information from the previous layers. In the forward phase, it enables the network to use the information from two complementary feature pyramids, bottom-up and top-down, without losing any of the information. In the backward phase, the gradient is distributed equally to all inputs, leading to efficient training. The element-wise maximum, which routes the gradient only to the higher input, provides a regularization effect to some extent, leading to unstable training and yielding the inferior performance in this experiment. The element-wise product approach, which can assign a large gradient to a small input and a small gradient to a large input, can complicate the convergence of the network. However in this case, because the two feature pyramids tend to be complementary, it enables the two feature pyramids to be balanced. Note that all four of these different implementations outperform the baseline in which only one pathway is used independently, showing that utilizing both pathways is crucial while the best-combining strategy boosts the performance further.

4.1.5 mAP in a high-recall range

The most commonly used evaluation technique for object detection is mAP. AP (average precision) refers to the concept of integrating precision as recall is varied from 0 to 1 and mAP is defined as the average of AP for all object classes. However, in most practical cases, it is more important to achieve high precision within a high-recall range rather than in a low-recall range [26]. We evaluate our model with mean average precision over Recall ≥ 0.7. Table 5 shows the results, indicating that even in the high-recall range our model achieves high precision and outperforms SSD and StairNet [46] (i.e., SSD + top-down pathway) significantly. This indicates that our method is far more practical than other methods. Note that the APs at a recall of 1 are 0 for all cases. This is due to the fact that we cannot recall all of the objects in the test images regardless of how much we lower the score threshold.

4.1.6 Robust to reduced parameters

We also explore robustness of our network against changes in the total number of parameters. We reduce the fixed channel dimension by a factor of 2, starting from the default value 256. As shown in Table 6, the accuracy drops off gently. Note that even when we push the model to extreme cases, using 32 channel dimensions overall in network, our model still outperforms the baseline SSD, which uses more parameters (23.6M vs 26.5M). This implies that GBFPN utilizes the feature pyramid very well even with a limited capacity and produces a finer feature pyramid. This motivates us to apply our GBFPN to a lightweight SSD model which adopts MobileNet [22] as its backbone network.

Table 5 mAP at Recall ≥ 0.7: mean average precision over a recall ≥ 0.7

Method	Data	Recall 0.7	0.8	0.9	1	mAP@0.7+	mAP
SSD300*[33]	07 + 12	81.3	67.1	37.6	0	46.5	77.8
StairNet [46]	07 + 12	83.5	70.1	39.8	0	48.4	79.0
GBFPN-SSD300	07 + 12	84.6	72.6	45.6	0	50.7	79.9

In most practical cases, it is more important to achieve high precision within a high-recall range rather than a low-recall range. Our GBFPN-SSD300 outperforms SSD and StairNet significantly in this region. *Tested in voc2007 test*

Table 6 Robust to reduced capacity

Inter channels	Params	mAP
256	33.7M	79.9
128	26.7M	79.4
64	24.4M	79.3
32	23.6M	78.9
Base (SSD300*)	26.5M	77.8

Our method GBFPN is robust against to the severe reduction of model parameters. Note that even we use far fewer parameters than the baseline SSD, performance of the proposed method is still meaningfully higher. *Tested in voc2007 test*

Table 7 Using MobileNet backbone

Backbone network	Detector	Params	mAP
MobileNet	SSD300*	5.8M	68.1
MobileNet	StairNet300	6.1M	69.9
MobileNet	GBFPN-SSD300	6.1M	70.7

We have shown that our method (GBFPN) is robust to parameter reduction, demonstrating its great potential for low-end devices. We can clearly note that given a lightweight backbone network, the GBFPN-SSD method shows significantly better performance than the baselines, showing its efficacy again in this limited-setting as well. *Tested in voc2007 test*

4.1.7 Using a lightweight backbone network

We evaluate the potential ability of GBFPN for low-end devices. For this experiment, we reduce the channel dimension from 256 to 128 and apply it on top of SSD. Table 6 shows that GBFPN significantly improves the performance with using only a few parameters overall. We also compare the GBFPN method with StairNet [46] in the lightweight setting (Table 7). We use similar numbers of parameters for in both methods, thus avoiding a performance improvement by simply increasing the capacity. We clearly note that GBFPN outperforms StairNet again even with limited parameters (i.e., 6.1M).

4.1.8 Comparison with state-of-the-art models

Table 8 shows our results on the PASCAL VOC 2007 test set. We observe that GBFPN-SSD achieves state-of-the-art results for input image sizes of both 300 and 512. SSD300* is the reproduced version in the PyTorch framework, and we achieved a value of 77.8 in this case. GBFPN-SSD300 achieves a mAP of 79.9, outperforming SSD by a large margin. Our model even outperforms DSSD321 [14] and BlitzNet300 [11], which rely on the deeper backbone networks ResNet101 and ResNet50, respectively. Because DSSD also adopts a StairNet-style top-down enrichment network, we can infer that GBFPN is a better solution to construct an enhanced feature pyramid. Moreover, due to its heavy design, these earlier method adopt multi-stage training schedules, while GBFPN-SSD is an end-to-end trainable model. BlitzNet uses extra segmentation labels for training, while our method only uses standard detection labels. The proposed method shows comparable performance with the approach of Zhang et al. [63], which introduces anchor refinement process in SSD. We believe our method can be further improved by adopting this mechanism. For the larger input image size of 512, we observe a similar phenomenon. GBFPN-SSD512 performs 81.8 mAP, outperforming DSSD 513 and BlitzNet 512 again as well. It is important to note that the original inference speed of SSD has significantly improved by adopting PyTorch 0.2.0 and cuDNN v6. Thus, we inherit the advantages of these newest versions for a rapid inference speed.

4.2 MS COCO

To validate the proposed GBFPN further, we conduct experiments on the MS COCO dataset. Following earlier work [14,33], we use the train/val35k set for training. We set the batch size to 32. We keep the original SSD strategy, which decreases the size of the default boxes, as objects in COCO tend to be smaller than those in PASCAL VOC. At the beginning of training, we apply the warm-up technique that gradually increases the learning rate from 10^{-6} to 4×10^{-3} during the first five epochs and then decrease it after 80 and 100 epochs by a factor of 10, ending up at 140.

4.2.1 Comparison with state-of-the-art models

Table 9 shows our results on MS COCO test-dev 2015. Note that GBFPN-SSD300 and GBFPN-SSD512 achieve

Table 8 PASCAL VOC 2007 *test* detection results

Method	Training data	Backbone network	mAP	fps
YOLO [41]	07 + 12	GoogLeNet	63.4	45
YOLOV2_352 [40]	07 + 12	DarkNet-19	73.7	81
SSD300* [33]	07 + 12	VGGNet	77.8	125
SSD300 [33]	07 + 12	VGGNet	77.5	62
SSD321 [14]	07 + 12	ResNet101	77.1	11.2
DSSD321 [14]	07 + 12	ResNet101	78.6	9.5
BlitzNet300 (s4) [11]	07 + 12 + *s*	ResNet50	79.1	24
RSSD300 [26]	07 + 12	VGGNet	78.5	35
DSOD300 [48]	07 + 12	DS/64-192-48-1	77.7	17.4
ASIF-Det320 [7]	07 + 12	VGGNet	79.2	33
RefineDet320 [63]	07 + 12	VGGNet	80.0	40.3
GBFPN-SSD300	07 + 12	VGGNet	79.9	83**
YOLOV2_544 [40]	07 + 12	DarkNet-19	78.6	40
SSD512 [33]	07 + 12	VGGNet	79.8	26
SSD512 [14]	07 + 12	ResNet101	80.6	6.8
DSSD513 [14]	07 + 12	ResNet101	81.5	5.5
BlitzNet512 (s8) [11]	07 + 12 + *s*	ResNet50	81.5	19.5
RSSD512 [26]	07 + 12	VGGNet	80.8	16.6
RefineDet512 [63]	07 + 12	VGGNet	81.8	24.1
GBFPN-SSD512	07 + 12	VGGNet	81.8	56**

07 + 12: 07 trainval + 12 trainval. 07 + 12 + *S*: 07 + 12 plus segmentation labels. We use GTX-1080ti along with the newest PyTorch version of 0.2.0 and cuDNN version 6 to compute inference speed. This improves original inference time of SSD by a factor of 2 (62 fps → 125 fps)

Table 9 MS COCO *test-dev2015* detection results

Method	Training data	Backbone network	mAP(0.5:0.95)
YOLOV2 [40]	trainval35k	DarkNet	21.6
SSD300 [33]	trainval35k	VGGNet	25.1
SSD321 [14]	trainval35k	ResNet101	28.0
DSSD321 [14]	trainval35k	ResNet101	28.0
ASIF-Det320 [7]	trainval35k	VGGNet	28.1
RefineDet320 [63]	trainval35k	VGGNet	29.4
GBFPN-SSD300	trainval35k	VGGNet	28.5
SSD512 [33]	trainval35k	VGGNet	28.8
SSD513 [14]	trainval35k	ResNet101	31.2
DSSD513 [14]	trainval35k	ResNet101	33.2
RefineDet512 [63]	trainval35k	VGGNet	33.0
GBFPN-SSD512	trainval35k	VGGNet	33.0

Note that accuracy improvement of our method only relies on the GBFPN, while other methods heavily depend on deep backbone networks along with the multi-stage training. Taking this into account, GBFPN-SSD method achieves comparable performance. We believe adopting deep backbone network or tuning the hyper-parameter extensively can bring an extra performance boost

mAP of 28.5% and 33.0%, respectively. Regarding the larger model, the result of GBFPN-SSD512 is slightly inferior to but still comparable to the state-of-the-art models. Because our method only relies on GBFPN, using lightweight VGGNet, the absolute value of the accuracy can be pushed by adopting a deep backbone network [18,57] or by conducting heavy hyper-parameter tuning. We can also consider using the recently proposed loss function tuning strategy [32]. However, because these methods are not the focus of this paper, we remain with VGGNet-based SSD and demonstrate its improvement over the baseline. Similar to the previous experiments shown in Table 8, we confirm that placing GBFPN on top of the SSD is an effective solution to enhance the pure

feature pyramid method, demonstrating its effectiveness with a large-scale challenging dataset [30] as well.

5 Conclusions

We have presented what is termed the gated bidirectional feature pyramid network (GBFPN), a new multi-scale aggregation method which improves detection outcomes. In contrast to previous methods, we explored a bottom-up pathway and a gating mechanism. The proposed GBFPN computes the final feature pyramid using two pathways with gating modules. To verify the efficacy of this approach, we conducted extensive experiments and confirmed that GBFPN is a better solution than the previous methods (i.e., SSD [33], StairNet [46]). Moreover, GBFPN-SSD achieves state-of-the-art results without bells and whistles. To ensure a clearer understanding of its process, we visualize how each pathway acts on the intermediate features. Interestingly, we observed that the two pathways play different roles and complement each other to produce the final enhanced feature pyramid. We believe that our findings provide a practical solution for other vision research areas and applications as well.

References

1. Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J., Ogden, J.M.: Pyramid methods in image processing. RCA Eng. **29**(6), 33–41 (1984)
2. Bell, S., Lawrence Zitnick, C., Bala, K., Girshick, R.: Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2016)
3. Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: Proceedings of European Conference on Computer Vision (ECCV), Springer (2016)
4. Chen, K., Wang, J., Chen, L.C., Gao, H., Xu, W., Nevatia, R.: Abc-cnn: an attention based convolutional neural network for visual question answering. arXiv preprint arXiv:1511.05960 (2015)
5. Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., Chua, T.S.: Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning (2017)
6. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In: IEEE Transactions Pattern Analysis and Machine Intelligence (TPAMI), IEEE (2018a)
7. Chen, X., Li, W., Wu, Q., Meng, F.: Adaptive multi-scale information flow for object detection (2018b)
8. Corbetta, M., Shulman, G.L.: Control of goal-directed and stimulus-driven attention in the brain. Nat. Rev. Neurosci. **3**(3), 201 (2002)
9. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: object detection via region-based fully convolutional networks. In: Proceedings of Neural Information Processing Systems (NIPS), pp. 379–387 (2016)
10. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), IEEE (2005)
11. Dvornik, N., Shmelkov, K., Mairal, J., Schmid, C.: Blitznet: a real-time deep network for scene understanding. In: Proceedings of International Conference on Computer Vision (ICCV) (2017)
12. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The Pascal visual object classes (voc) challenge. Int. J. Comput. Vis. **88**(2), 303–38 (2010)
13. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), IEEE (2010)
14. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659 (2017)
15. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2014)
16. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2015)
17. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: European Conference on Computer Vision, Springer, pp 346–361 (2014)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2016)
19. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of International Conference on Computer Vision (ICCV), IEEE (2017a)
20. He, P., Huang, W., He, T., Zhu, Q., Qiao, Y., Li, X.: Single shot text detector with regional attention. In: The IEEE International Conference on Computer Vision (ICCV) (2017b)
21. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
22. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
23. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks (2017)
24. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2017)
25. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (1998)
26. Jeong, J., Park, H., Kwak, N.: Enhancement of ssd by concatenating feature maps for object detection. Proceedings of British Machine Vision Conference (BMVC) (2017)
27. Kong, T., Yao, A., Chen, Y., Sun, F.: Hypernet: towards accurate region proposal generation and joint object detection. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2016)
28. Larochelle, H., Hinton, G.E.: Learning to combine foveal glimpses with a third-order Boltzmann machine. In: Proceedings of Neural Information Processing Systems (NIPS) (2010)

29. Li, H., Liu, Y., Ouyang, W., Wang, X.: Zoom out-and-in network with recursive training for object proposal. arXiv preprint arXiv:1702.05711 (2017)
30. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: common objects in context. In: Proceedings of European Conference on Computer Vision (ECCV), Springer (2014)
31. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2017a)
32. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection (2017b)
33. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Proceedings of European Conference on Computer Vision (ECCV), Springer (2016)
34. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2015)
35. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. (IJCV) **60**(2), 91–110 (2004)
36. Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Eleventh Annual Conference of the International Speech Communication Association (2010)
37. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Proceedings of European Conference on Computer Vision (ECCV), Springer (2016)
38. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of International Conference on Computer Vision (ICCV), pp 1520–1528 (2015)
39. Park, J., Woo, S., Lee, J.Y., Kweon, I.S.: Bam: bottleneck attention module (2018)
40. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2017)
41. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2016)
42. Ren, J., Chen, X., Liu, J., Sun, W., Pang, J., Yan, Q., Tai, Y.W., Xu, L.: Accurate single stage detector using recurrent rolling convolution. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), IEEE (2017)
43. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. In: Proceedings of Neural Information Processing Systems (NIPS), pp 91–99 (2015)
44. Rensink, R.A.: The dynamic representation of scenes. In: Visual cognition 7.1-3 (2000)
45. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer (2015)
46. Sanghyun, W., Soonmin, H., So, K.I.: Stairnet: top-down semantic aggregation for accurate one shot detection. In: Proceedings of Winter Conference on Applications of Computer Vision (WACV) (2018)
47. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks (2014)
48. Shen, Z., Liu, Z., Li, J., Jiang, YG., Chen, Y., Xue, X.: Dsod: Learning deeply supervised object detectors from scratch. In: Proceedings of International Conference on Computer Vision (ICCV) (2017)
49. Shrivastava, A., Sukthankar, R., Malik, J., Gupta, A.: Beyond skip connections: top-down modulation for object detection. arXiv preprint arXiv:1612.06851 (2016)
50. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., et al.: Going deeper with convolutions. Cvpr (2015)
51. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of Association for the Advancement of Artificial Intelligence (AAAI) (2017)
52. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective Search for Object Recognition. Springer, Berlin (2013)
53. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification (2017a)
54. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. arXiv preprint arXiv:1711.07971 (2017b)
55. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of European Conference on Computer Vision (ECCV) (2018)
56. Xiang, W., Zhang, D.Q., Athitsos, V., Yu, H.: Context-aware single-shot detector. arXiv preprint arXiv:1707.08682 (2017)
57. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), IEEE (2017)
58. Xu, H., Saenko, K.: Ask, attend and answer: exploring question-guided spatial attention for visual question answering. In: Proceedings of European Conference on Computer Vision (ECCV), Springer (2016)
59. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: neural image caption generation with visual attention (2015)
60. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.: Stacked attention networks for image question answering. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2016)
61. You, Q., Jin, H., Wang, Z., Fang, C., Luo, J.: Image captioning with semantic attention. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2016)
62. Zeng, X., Ouyang, W., Yan, J., Li, H., Xiao, T., Wang, K., Liu, Y., Zhou, Y., Yang, B., Wang, Z., et al.: Crafting gbd-net for object detection. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **40**(9), 2109–2123 (2017)
63. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection (2018)
64. Zhou, H., Li, Z., Ning, C., Tang, J.: Cad: Scale invariant framework for real-time object detection. In: Proceedings of International Conference on Computer Vision (ICCV) (2017)

Sanghyun Woo is a Ph.D. student in Electrical Engineering Department of Korea Advanced Institute of Science and Technology (KAIST), South Korea. He received his B.S. degree and M.S. degree in Electrical Engineering from Seoul National University (SNU) and KAIST, in 2017 and 2019, respectively. His research interests include deep learning, object recognition, and image/video processing. He is a student member of the IEEE.

Soonmin Hwang received the B.S. degree in Electronics and Computer Engineering from Hanyang University, Seoul, South Korea, in 2012, and the M.S. degree and Ph.D. degree in Electrical Engineering from KAIST, Daejeon, South Korea, in 2014 and 2019, respectively. He was an intern in computer vision group at Qualcomm Research Austria, Vienna, in 2016. He was awarded Gold prize (1st place in signal processing) in Samsung HumanTech paper award in 2017. His research interest includes robust visual perception for autonomous driving using deep learning and sensor fusion, especially under harsh environments such as nighttime and sensor fault.

Ho-Deok Jang received his B.S. degree in Electrical Engineering from Hongik University in 2017 and M.S. degree in Electrical Engineering (Division of Future Vehicle) from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2019. His research interests include deep learning-based visual recognitions such as object detection and segmentation.

In So Kweon is a professor in EE Department of KAIST, South Korea. He received the B.S. and M.S. degrees in mechanical design and production engineering from Seoul National University, Seoul, Korea, in 1981 and 1983, respectively, and the Ph.D. degree in robotics from the Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, in 1990. He worked for the Toshiba R&D Center, Japan, and joined the Department of Automation and Design Engineering, KAIST, Seoul, Korea, in 1992, where he is now a professor with the Department of Electrical Engineering. He is a recipient of the best student paper runner-up award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 09). His research interests include camera and 3D sensor fusion, color modeling and analysis, visual tracking, and visual SLAM. He was the program co-chair for the Asian Conference on Computer Vision (ACCV 07) and was the general chair for the ACCV 12. He is also on the editorial board of the International Journal of Computer Vision. He is a member of the IEEE and the KROS.