Crafting GBD-Net for Object Detection

Xingyu Zeng*,Wanli Ouyang*,Junjie Yan, Hongsheng Li,Tong Xiao, Kun Wang, Yu Liu, Yucong Zhou, Bin Yang, Zhe Wang,Hui Zhou, Xiaogang Wang,

***Abstract*—The visual cues from multiple support regions of different sizes and resolutions are complementary in classifying a candidate box in object detection. Effective integration of local and contextual visual cues from these regions has become a fundamental problem in object detection. In this paper, we propose a gated bi-directional CNN (GBD-Net) to pass messages among features from different support regions during both feature learning and feature extraction. Such message passing can be implemented through convolution between neighboring support regions in two directions and can be conducted in various layers. Therefore, local and contextual visual patterns can validate the existence of each other by learning their nonlinear relationships and their close interactions are modeled in a more complex way. It is also shown that message passing is not always helpful but dependent on individual samples. Gated functions are therefore needed to control message transmission, whose on-or-offs are controlled by extra visual evidence from the input sample. The effectiveness of GBD-Net is shown through experiments on three object detection datasets, ImageNet, Pascal VOC2007 and Microsoft COCO. Besides the GBD-Net, this paper also shows the details of our approach in winning the ImageNet object detection challenge of 2016, with source code provided on https://github.com/craftGBD/craftGBD. In this winning system, the modified GBD-Net, new pretraining scheme and better region proposal designs are provided. We also show the effectiveness of different network structures and existing techniques for object detection, such as multi-scale testing, left-right flip, bounding box voting, NMS, and context.**

***Index Terms*—Convolutional neural network, CNN, deep learning, deep model, object detection.**

I. Introduction

Object detection is one of the fundamental vision problems. It provides basic information for semantic understanding of images and videos. Therefore, it has attracted a lot of attentions [1], [2], [3]. In this paper, we regard detection as a problem of classifying candidate boxes. Due to large variations in viewpoints, poses, occlusions, lighting conditions and background, object detection is challenging. Recently, since the seminal work in [4], convolutional neural networks (CNNs) [5], [6], [7], [8], [9] have been proved to be effective for object detection because of its power in learning features.

In object detection, a candidate box is considered as true-positive for an object category if the intersection-over-union (IoU) between the candidate box and the ground-truth box is greater than a threshold. When a candidate box covers only a part of the ground-truth regions, there are some potential problems.

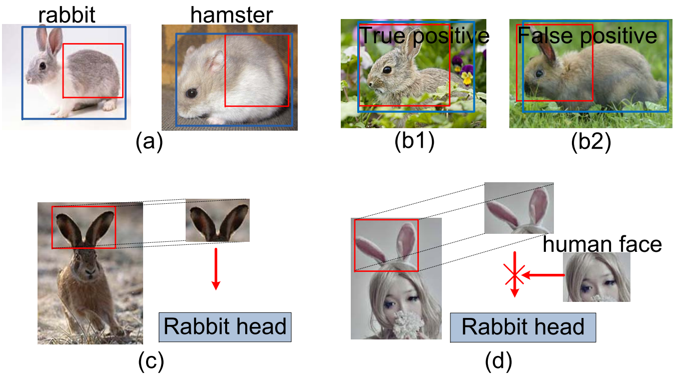


Fig. 1. The necessity of passing messages among features from supporting regions of different resolutions, and controlling message passing according different image instances. Blue windows indicate the ground truth bounding boxes. Red windows are candidate boxes. It is hard to classify candidate boxes which cover parts of objects because of similar local visual cues in (a) and variation of occlusion in (b). Local details of rabbit ears are useful for recognizing the rabbit head in (c). The contextual human head helps to find that the rabbit ear worn on human head should not be used to validate the existence of the rabbit head in (d). Best viewed in color.

- Visual cues in this candidate box may not be sufficient to distinguish object categories. Take the candidate boxes in Fig. 1(a) for example, they cover parts of animal bodies and have similar visual cues, but with different ground-truth class labels. It is hard to distinguish their class labels without information from larger surrounding regions of the candidate boxes.
- Classification on the candidate boxes depends on how much an object is occluded, which has to be inferred from larger surrounding regions. Because of occlusion, the candidate box covering a rabbit head in Fig. 1(b1) should be considered as a true positive of rabbit, because of large IoU with the ground truth. Without occlusion, however, the candidate box covering a rabbit head in Fig. 1(b2) should **not** be considered as a true positive because of small IoU with the ground truth.

To handle these problems, contextual regions surrounding candidate boxes are naturally helpful. Besides, surrounding regions also provide contextual information about background and other nearby objects to help detection. Therefore, in our deep model design and some existing works [10], [11], information from surrounding regions are used to improve classification of a candidate box.

On the other hand, when CNN takes a large region as input, it sacrifices the ability in describing local details, which are sometimes critical in discriminating object classes, since CNN

Xingyu Zeng (equal contribution), Wanli Ouyang (equal contribution), Tong Xiao, Kun Wang, Hongsheng Li, Zhe Wang, Hui Zhou and Xiaogang Wang are with the Department of Electronic Engineering at the Chinese University of Hong Kong, Hong Kong. Junjie Yan,Yu Liu, Yucong Zhou, Bin Yang are with the Sensetime Group Limited.

encodes input to a fixed-length feature vector. For example, the sizes and shapes of ears are critical details in discriminating rabbits from hamsters. But they may not be identified when they are in a very small part of the CNN input. It is desirable to have a network structure that takes both surrounding regions and local part regions into consideration. Besides, it is well-known that features from different resolutions are complementary [12].

One of our motivations is that features from different resolutions and support regions validate the existence of one another. For example, the existence of rabbit ears in a local region helps to strengthen the existence of a rabbit head, while the existence of the upper body of a rabbit in a larger contextual region also helps to validate the existence of a rabbit head. Therefore, we propose that features with different resolutions and support regions should pass messages to each other in multiple layers in order to validate their existences jointly during both feature learning and feature extraction. This is different from the naive way of learning a separate CNN for each support region and concatenating feature vectors or scores from different support regions for classification.

Our further motivation is that care should be taken when passing messages among contextual and local regions. The messages are not always useful. Taking Fig. 1(c) as an example, the local details of the rabbit ear is helpful in recognizing the rabbit head, and therefore, its existence has a large weight in determining the existence of the rabbit head. However, when this rabbit ear is artificial and worn on a girl's head in Fig. 1(d), it should not be used as the evidence to support the existence of a rabbit head. Extra information is needed to determine whether the message of a contextual visual pattern, e.g. rabbit ear, should be transmitted to support the existence of a target visual pattern, e.g. rabbit head. In Fig. 1(d), for example, the extra human-face visual cues indicates that the message of the rabbit ear should not be transmitted to strengthen the evidence of seeing the rabbit head. Taking this observation into account, we design a network that uses extra information from the input image region to adaptively control message transmission.

The ILSVRC16 challenge and COCO16 challenge for object detection has many entries that are worse than the best of the last year, although many of them have already used the same network structure or better network structures compared with the champion of the last year. The reason is from the details on implementing the object detection system. Using the ImageNet dataset and our winner system, we provide ablation study on the effectiveness of recent techniques for object detection, such as network structure, multi-scale testing, left-right flip, bounding box voting, NMS, context, and model ensemble. The code for these techniques is also provided online.

In this paper, we propose a gated bi-directional CNN (GBD-Net) architecture that adaptively models interactions of contextual and local visual cues during feature learning and feature extraction. Our contributions are in three-fold.

- A bi-directional network structure is proposed to pass messages among features from multiple support regions of different resolutions. With this design, local patterns pass detailed visual messages to larger patterns and large patterns passes contextual visual messages in both directions. Therefore, local and contextual features cooperate with each other on improving detection accuracy. We implement message passing by convolution.
- We propose to control message passing with gate functions. With such gate functions, message from a found pattern is transmitted only when it is useful for some samples, but is blocked for others.
- A new deep learning pipeline for object detection. It effectively integrates region proposal, feature representation learning, context modeling, and model averaging into the detection system. Detailed component-wise analysis is provided through extensive experimental evaluation. This paper also investigates the influence of CNN structures for the large-scale object detection task under the same setting. The details of our submission to the ImageNet Object Detection Challenge is provided in this paper, with source code provided online.

The proposed GBD-Net is implemented under the fast RCNN detection framework [13]. The effectiveness is validated through the experiments on three datasets, ImageNet [2], PASCAL VOC2007 [1] and Microsoft COCO [3].

An earlier version of this paper is published in [14]. In this journal version, we modify the GBD-Net, which is shown to improve mAp by 1.1%, compared with the original GBD-Net in [14]. A better pretraining approach adapting for the roi-pooling operation in the Fast-RCNN framework is also added, which improves mAP by 0.8%. Details on our approach in the ImageNet challenge of 2016 are also provided. We show step-by-step how the existing techniques are combined with the modified GBD-Net to reach the best accuracy in the ImageNet Challenge of 2016.

II. Related work

Impressive improvements have been achieved on object detection in recent years. They mainly come from better region proposals, detection pipeline, feature learning algorithms and CNN structures, iterative bounding box regression and making better use of local and contextual visual cues.

Region proposal. Selective search [15] obtained region proposals by hierarchically grouping segmentation results. Edgeboxes [16] counted the number of contours enclosed by a bounding box for inferring the likelihood of an object. Faster RCNN [17] and CRAFT [18] obtained region proposals with the help of convolutional neural networks. Pont-Tuest and Van Gool [19] studied the statistical difference between the Pascal-VOC dataset [1] and Microsoft CoCo dataset [3] to obtain better object proposals. In this paper, we adopt an improved version of the CRAFT in providing the region proposals.

Iterative regression. Since the candidate regions are not very accurate in locating objects, fine-grained search [20], multi-region CNN [11], LocNet [21] and AttractioNet [22] were proposed for more accurate localization of objects. These approaches conducted bounding box regression iteratively so that the candidate regions gradually move towards the ground truth object.

Object detection pipeline. The state-of-the-art deep learning based object detection pipeline RCNN [4] extracted CNN features from the warped image regions and applied a linear SVM as the classifier. By pre-training on the ImageNet classification dataset and finetuning on the target object detection dataset, it achieved great improvement in detection accuracy compared with previous sliding-window approaches that used handcrafted features on PASCAL-VOC and the large-scale ImageNet object detection dataset. In order to obtain a higher speed, Fast RCNN [13] shared the computational cost among candidate boxes in the same image and proposed a novel roi-pooling operation to extract feature vectors for each region proposal. Faster RCNN [17] combined the region proposal step with the region classification step by sharing the same convolution layers for both tasks. Region proposal is not necessary. Some recent approaches, e.g. Deep MultiBox [23], YOLO [24] and SSD [25], directly estimated the object classes from predefined sliding windows.

Learning and design of CNN. A large number of works [6], [7], [8], [9], [26], [27], [28], [29] aimed at designing network structures and they were found to be effective in the detection task. The works in [6], [7], [8], [9], [27] proposed deeper networks. People [30], [8], [31], [32] also investigated how to effectively train deep networks. Simonyan *et al.* [8] learn deeper networks based on the parameters in shallow networks. Ioffe *et al.* [30] normalized each layer inputs for each training mini-batch in order to avoid internal covariate shift. He *et al.* [31] investigated parameter initialization approaches and proposed parameterized RELU. Li *et al.* [33] proposed multi-bias non-linear activation (MBA) layer to explore the information hidden in the magnitudes of responses. Ouyang *et al.* investigate the use of attributes for learning better features [34]. The factor of long tail distribution in the number of samples for different classes is investigated in [34].

Our contributions focus on a novel bi-directional network structure to effectively make use of multi-scale and multi-context regions. Our design is complementary to above region proposals, pipelines, CNN layer designs, and training approaches. There are many works on using visual cues from object parts [35], [11], [36] and contextual information [35], [11], [37]. Gidaris *et al.* [11] adopted a multi-region CNN model and manually selected multiple image regions. Girshick *et al.* [36] and Ouyang *et al.* [35] learned the deformable parts from CNNs. In order to use the contextual information, multiple image regions surrounding the candidate box were cropped in [11], whole-image classification scores were used in [35]. These works simply concatenated features or scores from object parts or context while we pass message among features representing local and contextual visual patterns so that they validate the existence of each other by non-linear relationship learning. Experimental results show that GBD-Net is complementary to the approaches in [11], [35]. As a step further, we propose to use gate functions for controlling message passing, which was not investigated in existing works.

Passing messages and gate functions. Message passing at the feature level was studied using Recurrent neural network (RNN) for features of the same resolution [10] and gate functions are used to control message passing in long short-term memory (LSTM) networks [38]. However, both techniques have not been used to for features from different resolution and context yet, which is fundamental in object detection. Our message passing mechanism and gate functions are specifically designed for this setting. GBD-Net is also different from RNN and LSTM in the sense that it does not have recurrence and does not share parameters across resolutions/contexts.

III. Gated Bi-directional CNN

We briefly introduce the fast RCNN pipeline in Section III-A and then provide an overview of our approach in Section III-B. The use of roi-pooling for obtaining features of different resolutions and contexts is discussed in Section III-C. Section III-D focuses on the proposed bi-directional network structure and its gate function. Section III-E introduces the modified GBD structure. Section III-F explains the details of the training scheme.

A. Fast RCNN pipeline

We adopt the Fast RCNN[13] as the object detection pipeline with four steps.

- Step 1) Candidate box generation. Thousands or hundreds of candidate boxes are selected from a large pool of boxes.
- Step 2) Feature map generation. Given an image as the input of CNN, feature maps are generated.
- Step 3) Roi-pooling. Each candidate box is considered as a region-of-interest (roi) and a pooling function is operated on the CNN feature maps generated in the step 2. After roi-pooling, candidate boxes of different sizes are pooled to have the same feature size.
- Step 4) Classification. CNN features after roi-pooling go through several convolutions, pooling and fully connected layers to predict the class label and location refinement of candidate boxes.

B. Framework overview

An overview of the GBD-Net is shown in Fig. 2. Based on the fast RCNN pipeline, our proposed model takes an image as input, uses roi-pooling operations to obtain features with different resolutions and different support regions for each candidate box, and then the gated bi-direction layer is used for passing messages among features, and finally classification and bounding box regression are done. We use the Inception-v2 [30] as the baseline network structure, i.e. if only one support region and one branch is considered, Fig. 2 becomes a Inception-v2. Currently, messages are passed between features in one layer. It can be extended by adding more layers between the features of different resolutions for passing messages in these layers.

We use the same candidate box generation and feature map generation steps as the fast RCNN introduced in Section III-A. In order to take advantage of complementary visual cues in the surrounding/inner regions, the major modifications of fast RCNN are as follows.

- In the roi-pooling step, regions with the same center location but different sizes are pooled from the same feature maps for a single candidate box. The regions with

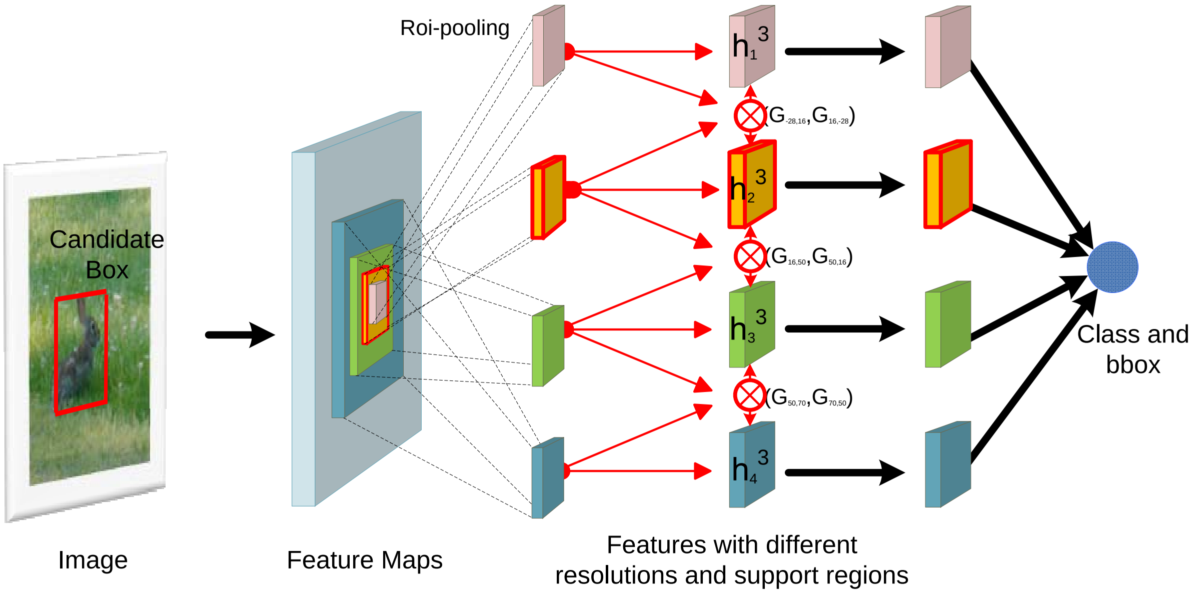


Fig. 2. Overview of our framework. The network takes an image as input and produces feature maps. The roi-pooling is done on feature maps to obtain features with different resolutions and support regions. Red arrows denote our gated bi-directional structure for passing messages among features. Gate functions G are defined for controlling the message passing rate. Then all features go through multiple CNN layers with shared parameters to obtain the final features that are used to predict the class and location refinement of bounding box. Using only $\mathbf{h}_2^3$ would reduce the network to Fast-RCNN. Parameters on black arrows are shared across branches, while parameters on red arrows are not shared. Best viewed in color.

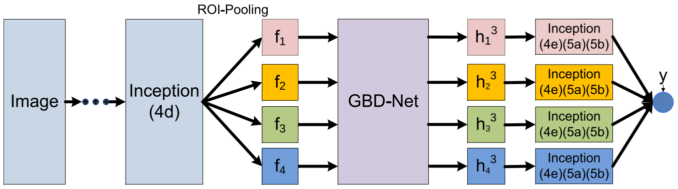


Fig. 3. Exemplar implementation of our model. The gated bi-directional network, dedicated as GBD-Net, is placed between Inception (4d) and Inception (4e). Inception (4e), (5a) and (5b) are shared among all branches.

different sizes before roi-pooling have the same size after roi-pooling. In this way, the pooled features correspond to different support regions and have different resolutions.

- Features with different resolutions optionally go through several CNN layers to extract their high-level features.
- The bi-directional structure is designed to pass messages among the roi-pooled features with different resolutions and support regions. In this way, features corresponding to different resolutions and support regions verify each other by passing messages to each other.
- Gate functions are used to control message transmission.
- After message passing, the features for different resolutions and support regions are then passed through several CNN layers for classification.

An exemplar implementation of our model is shown in Fig. 3. There are 9 inception modules in the Inception-v2 [30]. Roi-pooling of multiple resolutions and support regions is conducted after the 6th inception module, which is inception (4d). Then the gated bi-directional network is used for passing messages among features and outputs $\mathbf{h}_1^3$, $\mathbf{h}_2^3$, $\mathbf{h}_3^3$, and $\mathbf{h}_4^3$. After message passing, $\mathbf{h}_1^3$, $\mathbf{h}_2^3$, $\mathbf{h}_3^3$ and $\mathbf{h}_4^3$ are fed forward to the 7th, 8th, 9th inception modules and the average pooling layers separately and then used for classification. There is option to place roi-pooling and GBD-Net after different layers of the Inception-v2. In Fig. 3, they are placed after inception (4d).

C. Roi-pooling of features with different resolutions and support regions

The roi-pooling layer designed in [13] is used to obtain features with different resolutions and support regions. Given a candidate box $\mathbf{b}^o = [x^o, y^o, w^o, h^o]$ with center location (x^o, y^o), width w^o and height h^o, its padded bounding box is denoted by $\mathbf{b}^p$. $\mathbf{b}^p$ is obtained by enlarging the original box $\mathbf{b}^o$ along both x and y directions with scale p as follows:

$$\mathbf{b}^p = [x^o, y^o, (1+p)w^o, (1+p)h^o]\,. \tag{1}$$

In RCNN [4], p is 0.2 by default and the input to CNN is obtained by warping all the pixels in the enlarged bounding box $\mathbf{b}^p$ to a fixed size $w \times h$, where $w = h = 224$ for the Inception-v2 [30]. In fast RCNN [13], warping is done on feature maps instead of pixels. For a box $\mathbf{b}^o$, its corresponding feature box $\mathbf{b}^f$ on the feature maps is calculated and roi-pooling uses max pooling to convert the features in $\mathbf{b}^f$ to feature maps with a fixed size.

In our implementation, a set of padded bounding boxes $\{\mathbf{b}^p\}$ with $p = -0.2, 0.2, 0.8$, or 1.7 are generated for each candidate box $\mathbf{b}^o$. By roi-pooling on the CNN features, these boxes are warped into the same size, which is $14 \times 14 \times 608$ for Inception-v2. The CNN features of these padded boxes have different resolutions and support regions. In the roi-pooling step, regions corresponding to $\mathbf{b}^{-0.2}, \mathbf{b}^{0.2}, \mathbf{b}^{0.8}$ and $\mathbf{b}^{1.7}$ are

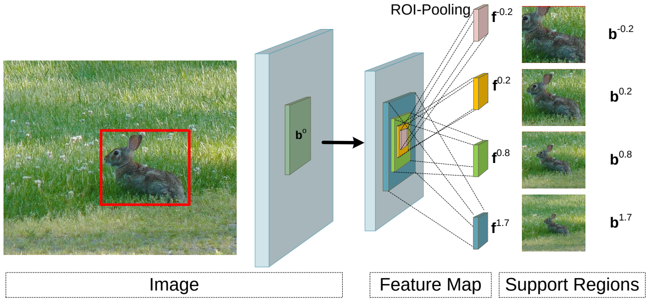


Fig. 4. Illustration of using roi-pooling to obtain CNN features with different resolutions and support regions. The red rectangle in the left image is a candidate box. The right four image patches show the supporting regions for $\{\mathbf{b}^p\}$. Best viewed in color.

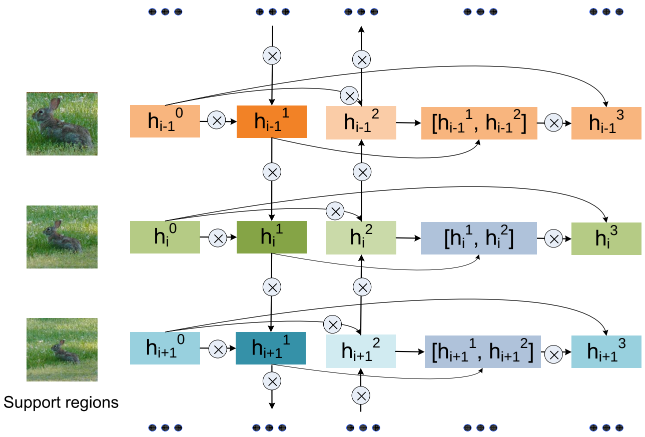


Fig. 5. Details of our bi-directional structure. $\otimes$ denotes convolution. The input of this structure is the features $\{\mathbf{h}_i^0\}$ of multiple resolutions and contextual regions. Then bi-directional connections among these features are used for passing messages across resolutions/contexts. The output $\mathbf{h}_i^3$ are updated features for different resolutions/contexts after message passing.

warped into features $\mathbf{f}^{-0.2}, \mathbf{f}^{0.2}, \mathbf{f}^{0.8}$ and $\mathbf{f}^{1.7}$ respectively. Figure 4 illustrates this procedure.

Since features $\mathbf{f}^{-0.2}, \mathbf{f}^{0.2}, \mathbf{f}^{0.8}$ and $\mathbf{f}^{1.7}$ after roi-pooling are of the same size, the context scale value p determines both the amount of padded context and also the resolution of the features. A larger p value means a lower resolution for the original box but more contextual information around the original box, while a small p means a higher resolution for the original box but less context.

D. Gated Bi-directional network structure (GBD-v1)

1) Bi-direction structure: Figure 5 shows the architecture of our proposed bi-directional network. It takes features $\mathbf{f}^{-0.2}, \mathbf{f}^{0.2}, \mathbf{f}^{0.8}$ and $\mathbf{f}^{1.7}$ as input and outputs features $\mathbf{h}_1^3$, $\mathbf{h}_2^3$, $\mathbf{h}_3^3$ and $\mathbf{h}_4^3$ for a single candidate box. In order to have features $\{\mathbf{h}_i^3\}$ with different resolutions and support regions cooperate with each other, this new structure builds two directional connections among them. One directional connection starts from features with the smallest region size and ends at features with the largest region size. The other is the opposite.

For a single candidate box $\mathbf{b}^o$, $\mathbf{h}_i^0 = \mathbf{f}^{p_i}$ representing features with context pad value p_i. The forward propagation for the proposed bi-directional structure can be summarized as follows:

$$\mathbf{h}_i^1 = \sigma(\mathbf{h}_i^0 \otimes \mathbf{w}_i^1 + \mathbf{b}_i^{0,1}) + \sigma(\mathbf{h}_{i-1}^1 \otimes \mathbf{w}_{i-1,i}^1 + \mathbf{b}_i^1), \quad (2)$$

(high res. to low pass)

$$\mathbf{h}_i^2 = \sigma(\mathbf{h}_i^0 \otimes \mathbf{w}_i^2 + \mathbf{b}_i^{0,2}) + \sigma(\mathbf{h}_{i+1}^2 \otimes \mathbf{w}_{i,i+1}^2 + \mathbf{b}_i^2), \quad (3)$$

(low res. to high pass)

$$\mathbf{h}_i^3 = \sigma(cat(\mathbf{h}_i^1, \mathbf{h}_i^2) \otimes \mathbf{w}_i^3 + \mathbf{b}_i^3), \quad (4)$$

(message integration)

- There are four different resolutions/contexts, $i = 1, 2, 3, 4$.
- $\mathbf{h}_i^1$ represents the updated features after receiving message from $\mathbf{h}_{i-1}^1$ with a higher resolution and a smaller support region. It is assumed that $\mathbf{h}_0^1 = 0$, since $\mathbf{h}_1^1$ has the smallest support region and receives no message.
- $\mathbf{h}_i^2$ represents the updated features after receiving message from $\mathbf{h}_{i+1}^2$ with a lower resolution and a larger support region. It is assumed that $\mathbf{h}_5^2 = 0$, since $\mathbf{h}_4^2$ has the largest support region and receives no message.
- $cat()$ concatenates CNN features maps along the channel direction.
- The features $\mathbf{h}_i^1$ and $\mathbf{h}_i^2$ after message passing are integrated into $\mathbf{h}_i^3$ using the convolutional filters $\mathbf{w}_i^3$.
- $\otimes$ represents the convolution operation. The biases and filters of convolutional layers are respectively denoted by $\mathbf{b}_*^*$ and $\mathbf{w}_*^*$.
- Element-wise RELU is used as the non-linear function $\sigma(\cdot)$.

From the equations above, the features in $\mathbf{h}_i^1$ receive the messages from the high-resolution/small-context features and the features $\mathbf{h}_i^2$ receive messages from the low-resolution/large-context features. Then $\mathbf{h}_i^3$ collects messages from both directions to have a better representation of the ith resolution/context. For example, the visual pattern of a rabbit ear is obtained from features with a higher resolution and a smaller support region, and its existence (high responses in these features) can be used for validating the existence of a rabbit head, which corresponds to features with a lower resolution and a larger support region. This corresponds to message passing from high resolution to low resolution in (2). Similarly, the existence of the rabbit head at the low resolution also helps to validate the existence of the rabbit ear at the high resolution by using (3). $\mathbf{w}_{i-1,i}^1$ and $\mathbf{w}_{i,i+1}^1$ are learned to control how strong the existence of a feature with one resolution/context influences the existence of a feature with another resolution/context. Even after bi-directional message passing, $\{\mathbf{h}_i^3\}$ are complementary and will be jointly used for classification in later layers.

Our bi-directional structure is different from the bi-direction recurrent neural network (RNN). RNN aims to capture dynamic temporal/spatial behavior with a directed cycle. It is assumed that parameters are shared among directed connections. Since our inputs differ in both resolutions and contextual regions, convolutions layers connecting them should learn different relationships at different resolution/context levels.

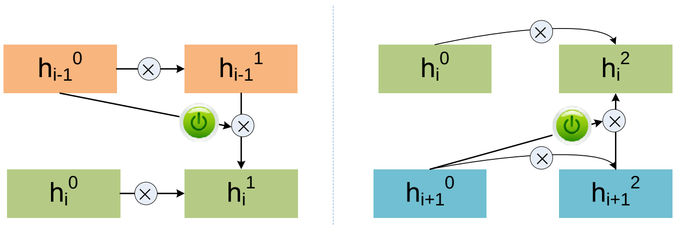


Fig. 6. Illustration of the bi-directional structure with gate functions. The $\otimes$ represents the convolution and the switch button represents the gate function. The left one corresponds to Eq. (5) and the right one corresponds to Eq. (6).

Therefore, the convolutional parameters for message passing are not shared in our bi-directional structure.

2) Gate functions for message passing: Instead of passing messages in the same way for all the candidate boxes, gate functions are introduced to adapt message passing for individual candidate boxes. Gate functions are also implemented as convolution. The design of gate function considers the following aspects.

- $\mathbf{h}_i^k$ has multiple feature channels. The learned gate filters are different for different channels.
- The message passing rates should be controlled by the responses to particular visual patterns which are captured by gate filters.
- The message passing rates can be determined by visual cues from nearby regions, e.g. in Fig. 1, a girl's face indicates that the rabbit ear is artificial and should not pass message to the rabbit head. Therefore, the size of gate filters should not be 1×1 and 3×3 is used in our implementation.

We design gate functions as convolution layers with the sigmoid non-linearity to make the message passing rate in the range of (0,1). With gate functions, message passing in (2) and (3) for the bi-directional structure is changed as follows:

$$\mathbf{h}_i^1 = \sigma(\mathbf{h}_i^0 \otimes \mathbf{w}_i^1 + \mathbf{b}_i^{0,1}) + G_i^1 \bullet \sigma(\mathbf{h}_{i-1}^1 \otimes \mathbf{w}_{i-1,i}^1 + \mathbf{b}_i^1), \tag{5}$$

$$\mathbf{h}_i^2 = \sigma(\mathbf{h}_i^0 \otimes \mathbf{w}_i^2 + \mathbf{b}_i^{0,2}) + G_i^2 \bullet \sigma(\mathbf{h}_{i+1}^2 \otimes \mathbf{w}_{i,i+1}^2 + \mathbf{b}_i^2), \tag{6}$$

$$G_i^1 = sigm(\mathbf{h}_{i-1}^0 \otimes \mathbf{w}_{i-1,i}^g + \mathbf{b}_{i-1,i}^g) \tag{7}$$

$$G_i^2 = sigm(\mathbf{h}_{i+1}^0 \otimes \mathbf{w}_{i+1,i}^g + \mathbf{b}_{i+1,i}^g) \tag{8}$$

where $sigm(\mathbf{x}) = 1/[1 + \exp(-\mathbf{x})]$ is the element-wise sigmoid function and $\bullet$ denotes element-wise product. G is the gate function to control message passing. It contains learnable convolutional parameters $\mathbf{w}_*^g, \mathbf{b}_*^g$ and uses features from the co-located regions to determine the rates of message passing. When G_i^* is 0, the message is not passed. The formulation for obtaining $\mathbf{h}_i^3$ is unchanged. Fig. 6 illustrates the bi-directional structure with gate functions.

E. The modified GBD structure (GBD-v2)

For the models submitted to ImageNet challenge, the GBD-v1 is modified. The modified GBD-Net structure has the following formulation:

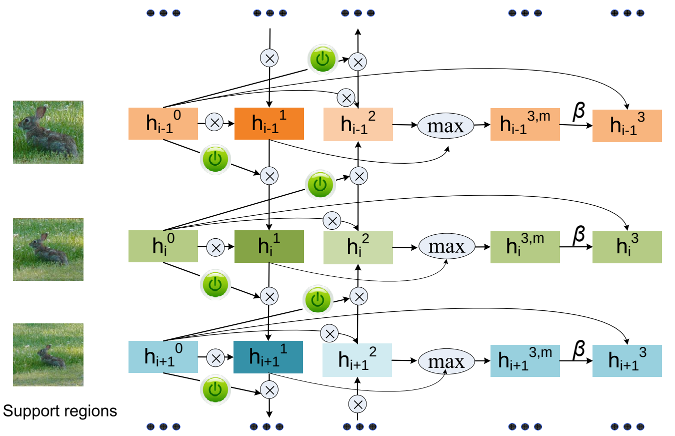


Fig. 7. Details of our modified bi-directional structure. Compared with the stucture in Fig. 5, an identity mapping layer is added from $\mathbf{h}_*^0$ to $\mathbf{h}_*^3$. The convolution from $[\mathbf{h}_*^1, \mathbf{h}_*^2]$ to $\mathbf{h}_*^3$ in Fig. 5 is changed into max-pooling.

$$\mathbf{h}_i^{3,m} = \max(\mathbf{h}_i^1, \mathbf{h}_i^2), \tag{9}$$

$$\mathbf{h}_i^3 = \mathbf{h}_i^0 + \beta \mathbf{h}_i^{3,m}, \tag{10}$$

where $\mathbf{h}_i^1$ and $\mathbf{h}_i^2$ are defined in (5) and (6). Fig. 7 shows the modified GBD structure. The operations requried for obtaining $\mathbf{h}_i^1$ and $\mathbf{h}_i^2$ are the same as before. The main changes are in obtaining $\mathbf{h}_i^3$. The changes made are as follows.

First, in the previous GBD structure, $\mathbf{h}_i^1$ and $\mathbf{h}_i^2$ are concatenated and then convoled by filters to produce the output $\mathbf{h}_i^3$, as shown in (4). In the modified sturcture, a max pooling is used for merging the information from $\mathbf{h}_i^1$ and $\mathbf{h}_i^2$. This saves the memory and computation required by the convolution in the previous GBD structure.

Second, we add an identity mapping layer in the struture, which corresoponds to the $\mathbf{h}_i^3 = \mathbf{h}_i^0 + ...$ in (4) and (10). The aim of the GBD structure is to refine the input $\mathbf{h}_i^0$ by using the messages from other contextual features. Since the parameters for the layers after the output $\mathbf{h}_i^3$ are from pretrained model, a drastic change of the output $\mathbf{h}_i^3$ from the input $\mathbf{h}_i^0$ would cause difficulty in training the layers after the layer $\mathbf{h}_i^3$ to adapt at the training stage. Therefore, this drastic change would lead to difficulty in learning a good model. When we train the previous GBD structure, careful initialization of the convolution parameters and the gate functions has to be done in order to learn well. For example, we have to set the gate function close 0 and the convolution parameter close to identity mapping for initialization. With the identity mapping layer, however, a simple initialization using the approach in [39] works well.

Third, a constant β is multiplied with the merged messages $\mathbf{h}_i^{3,m}$ from other contextual regions. We empirically found that it improves detection accuracy by using β to control the magnitude of the messages from other contextual features.

F. Implementation details, training scheme, and loss function

For the state-of-the-art fast RCNN object detection framework, CNN is first pre-trained with the ImageNet image classification data, and then utilized as the initial point for fine-tuning the CNN to learn both object confidence scores s and bounding-box regression offsets t for each candidate box. Our proposed framework also follows this strategy and randomly initialize the filters in the gated bi-direction structure while the other layers are initialized from the pre-trained CNN. The final prediction on classification and bounding box regression is based on the representations $\mathbf{h}_i^3$ in Eq. (10). For a training sample with class label y and ground-truth bounding box offsets $\mathbf{v} = [v_1, v_2, v_3, v_4]$, the loss function of our framework is a summation of the cross-entropy loss for classification and the smoothed L_1 loss for bounding box regression as follows:

$$L(y, t_c, \mathbf{v}, t_v) = L_{cls}(y, t_c) + \lambda[y \geq 1]L_{loc}(\mathbf{v}, \mathbf{t}_v), \quad (11)$$

$$L_{cls}(y, t_c) = -\sum_c \delta(y, c) \log t_c, \quad (12)$$

$$L_{loc}(\mathbf{v}, \mathbf{t}_v) = \sum_{i=1}^{4} \text{smooth}_{L_1}(v_i - t_{v,i}), \quad (13)$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| \leq 1 \\ |x| - 0.5 & otherwise \end{cases}, \quad (14)$$

where the predicted classification probability for class c is denoted by t_c, and the predicted offset is denoted by $\mathbf{t}_v = [t_{v,1}, t_{v,2}, t_{v,3}, t_{v,4}]$, $\delta(y, c) = 1$ if $y = c$ and $\delta(y, c) = 0$ otherwise. $\lambda = 1$ in our implementation. Parameters in the networks are learned by back-propagation.

G. Discussion

The GBD structure is built upon the features of different resolutions and contexts. Its placement is independent of the place of roi-pooling. In an extreme setting, roi-pooling can be directly applied on raw pixels to obtain features of multiple resolutions and contexts, and in the meanwhile the GBD structure can be placed in the last convolution layer for message passing. In this implementation, fast RCNN is reduced to RCNN where multiple regions surrounding a candidate box are cropped from raw pixels instead of feature maps.

IV. The object detection framework on ImageNet 2016

In this section, we describe object detection framework used for our submission to the 2016 ImageNet Detection Challenge.

A. Overview at the testing stage

- Step 1) Candidate box generation. An improved version of CRAFT in [18] is used for generating the region proposal.
- Step 2) Box classification. The GBD-Net predicts the class of candidate boxes.
- Step 3) Average of multiple deep model outputs is used to improve the detection accuracy.
- Step 4) Postprocessing. The whole-image classification scores are used as contextual scores for refining the detection scores. The bounding box voting scheme in [11] is adopted for adjusting the box locations based on its neigbouring boxes.

B. Candidate box generation

We use two versions of object proposal. In early version of our method, we use the solutions published in [18] which is denoted as Craft-v2. In the final ImageNet submission, we further improve the results and denote it as Craft-v3. A brief review of Craft-v2 and details of Craft-v3 are described as follows.

1) Craft-V1 and V2: In Craft [18], the RPN [17] is extended to be a two-stage cascade structure, following the "divide and conquer " strategy in detection task. In the first stage, the standard RPN is used to generate about 300 proposals for each image, which is similar to the setting in [17]. While in the second stage, a two-category classifier is further used to distinguish objects from background. Specially in the paper, we use a two-category fast RCNN [13]. It provides fewer and better localized object proposals than the standard RPN. Craft-V1, which was used in our earlier version [14], and Craft-V2 can be found in our early paper [18]. Craft-V1 and Craft-V2 are only different in pre-training. Craft-V1 is pre-trained from 1000-class image classification, Craft-V2 is pre-trained from RPN [17].

2) Craft-v3: Compared with Craft-v2, the differences in Craft-v3 includes:

- Random crop is used in model training, to ensure objects in different scales are roughly trained equally.
- Multi-scale pyramid is used in model testing, in order to improve recall of small objects.
- The positive and negative samples in RPN training are balanced to be 1:1.
- LocNet [21] object proposals are added, which we found are complementary to the Craft based proposals.

Implementation details and experimental comparison can be found in Section V-E1.

C. Box classification with GBD-Net

The GBD-Net is used for predicting the object category of the given candidate boxes. The preliminary GBD-Net structure in [14] was based on the Inception-v2. In the challenge we make the following modifications:

- The baseline network is pretrained on ImageNet 1000-class data with object-centric labels without adapting to fas RCNN. In the challenge, we learn the baseline network with object-centric labels by adapting it to fas RCNN.
- A ResNet with 269 layers is used as the baseline model for the best performing GBD-Net.
- The structure of GBD-Net is changed from GBD-v1 to GBD-v2, with details in Section III-E.

D. Pretraining the baseline

1) The baseline ResNet-269 model: The network structure of baseline ResNet with 269 layers is shown in Fig. 8. Compared with the ResNet [27] with 152 layers, we simply increase the number of stacked blocks for conv3_x, conv4_x, and conv5_x. The basic blocks adopt the identity mapping used in [40]. At the pre-training stage, the stochastic depth in

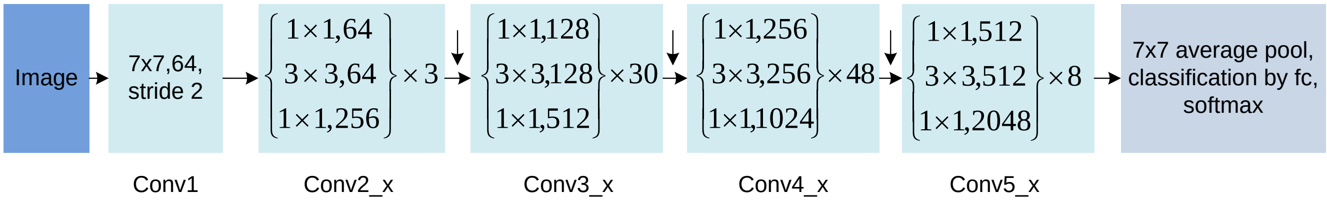


Image
7x7,64, stride 2
$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$
$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times30$
$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times48$
$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times8$
7x7 average pool, classification by fc, softmax
Conv1
Conv2_x
Conv3_x
Conv4_x
Conv5_x

Fig. 8. Architecture for the baseline ResNet-269. Building blocks are the identity mapping blocks used in [40], with the numbers of blocks stacked. Downsampling is performed by conv3_1, conv4_1, and conv5_1 with a stride of 2.

[41] is used. Stochastic depth is found to reduce training time and test error in [41]. For fast RCNN, we place the roi-pooling after the 234th layer, which is in the stacked blocks conv4_x.

2) Adapt the pretraining for roi-pooling: Pretraining can be done on the ImageNet 1000-class image classification data by taking the whole image as the input of CNN, this is called image-centric pretraining. On the other hand, since bounding box labels are provided for these classes in ImageNet, the input of CNN can be obtained from cropping and warping image patches in the bounding boxes, which is called object-centric pretraining. For the RCNN framework, it is found by our previous work [35] that object-centric pretraining performs better than image-centric pretraining.

For fast RCNN, however, we found that the CNN with object-centric pretraining does not perform better than the 1000-class image-centric pretraining. Take the Inception-v2 as an exmaple, after finetuning on the ImageNet train+val1 data, the Inception-v2 with image-centric pretraining has 49.1% mAP on the ImageNet val2 while the Inception-v2 with object-centric pretraining drops to 48.4% mAP. This drop in mAp is caused by the roi-pooling in fast RCNN. RCNN and Fast RCNN use different ways to obtain features of the same size for candidate regions of different sizes. RCNN warps the candidate image region. Fast RCNN keeps the image size unchanged and uses roi-pooling for warping features. Warping at image level for RCNN and roi-pooling at feature level are not equivalent operations.

In order to have the pretrained CNN aware of the difference mentioned above, we pretrain on object-centric task with roi-pooling layer included. Denote the bounding box of an object by (x_1, y_1, x_2, y_2). When pretraining the Inception-v2 with the roi-pooling layer, we include 32 pixels as the extrapolated region for this object. Therefore, the target box is $(x_{t,1}, y_{t,1}, x_{t,2}, y_{t,2}) = (x_1 - 32, y_1 - 32, x_2 + 32, y_2 + 32)$. To augment data, we randomly shake the target box as follows:

$$\begin{aligned}\mathbf{b}_f &= (x_{f,1}, y_{f,1}, x_{f,2}, y_{f,2}) \qquad (15)\\ &= (x_{t,1} + \alpha_1 W, y_{t,1} + \alpha_2 H, x_{t,2} + \alpha_3 W, y_{t,2} + \alpha_4 H),\end{aligned}$$

where $W = x_2 - x_1 + 1$ and $H = y_2 - y_1 + 1$ are respectively the width and height of the bounding box. α_1, α_2, α_3, and α_4 are randomly sampled from $[-0.1\ 0.1]$ independently. The image region within the box $\mathbf{b}_f$ in (15) is warped into an image with shorter side randomly sampled from $\{300, 350, 400, 450, 500\}$ and the longer side constrained to be no greater than 597. Batch size is set as 256 with other settings the same as Inception-v2. We observe 0.8% mAP gain for Inception-v2 with this pretraining when compared with pretraining by image-centric classification.

E. Technical details on improving performance

1) Multi-scale testing: Multi-scale training/testing has been developed in [13], [43] by selecting features from a feature pyramid. We only use the multi-scale input at the testing stage. With a trained model, we compute feature maps on an image pyramid, with the shorter side of the image being $\{400, 500, 600, 700, 800\}$ and longer size being no greater than 1000. Roi-pooling and its subsequent layers are performed on the feature map of one scale. We did not observe obvious improvement by averaging the scores of a bounding box using its features pooled from multiple scales.

2) Left-right flip: We adopt left-right flip at both the training and testing stages. At training stage, the training images are augmented by flipping them. The candidate boxes are flipped accordingly. At testing stage, an image and its corresponding candidate boxes are flipped and treated as the input of the CNN to obtained the detection scores for these boxes. The scores and estimated box locations from the original image and the flipped image are averaged.

3) Bounding box voting: The bounding box voting scheme in [11] is adopted. After finding the peaked box with the highest score on its neighborhood, the final object location is obtained by having each of the boxes that overlap with the peaked one by more than a threshold to vote for the bounding box location using its score as weight. This threshold is set as 0.5 for IoU.

4) Non-maximum suppression (NMS) threshold: For ImageNet, the NMS threshold was set as 0.3 by default. We empirically found 0.4 to be a better threshold. Setting this threshold from 0.3 to 0.4 provides 0.4-0.7% mAP gain, with variation for different models.

5) Global context: From the pretrained image-centric CNN model, we finetune on the ImageNet detection data by treating it as an image classification problem. The 200-class image classification score is then used for combining with the 200-class object detection scores by weighted averaging. The weights are obtained by greedy search from the val1 data.

6) Model ensemble: For model ensemble, 6 models are automatically selected by greedy search on ImageNet Det val2 from 10 models. The average of scores and bounding box regression results of these 6 models are used for obtaining the model averaging results.

TABLE I
OBJECT DETECTION MAP (%) ON IMAGENET VAL2 FOR STATE-OF-THE-ART APPROACHES WITH SINGLE MODEL (SGL) AND AVERAGED MODEL (AVG).

appraoch	RCNN [4]	Berkeley [4]	GoogleNet [9]	DeepID-Net[35]	Superpixel [42]	ResNet [27]	Ours
val2(sgl)	31.0	33.4	38. 5	48.2	42.8	60.5	65
val2(avg)	n/a	n/a	40.9	50.7	45.4	63.6	68

V. EXPERIMENTAL RESULTS

A. Implementation details

The GBD-net is implemented based on the fast RCNN pipeline. The Inception-v2 will be used for ablation study and our submission to the ILSVRC2016 is based on the ResNet with identity mapping [40] and the PolyNet [28]. The gated bi-directional structure is added after the 6th inception module (4d) of Inception-v2 and after the 234th layer for ResNet-269. In the GBD-Net, layers belonging to the baseline networks are initialized by these baseline networks pre-trained on the ImageNet 1000-class classification and localization dataset. The parameters in the GBD layers as shown in Fig. 5, which are not present in the pre-trained models, are randomly initialized when finetuning on the detection task. In our implementation of GBD-Net, the feature maps $\mathbf{h}_i^n$ for $n = 1, 2, 3$ in (2)-(4) have the same width, height and number of channels as the input $\mathbf{h}_i^0$ for $i = 1, 2, 3, 4$.

We evaluate our method on three public datasets, ImageNet object detection dataset [2], Pascal VOC 2007 dataset [1] and Microsoft COCO object detection dataset [3]. Since the ImageNet object detection task contains a sufficiently large number of images and object categories to reach a conclusion, evaluations on component analysis of our training method are conducted on this dataset. This dataset has 200 object categories and consists of three subsets. i.e., train, validation and test data. In order to have a fair comparison with other methods, we follow the same setting in [4] and split the whole validation subset into two subsets, val1 and val2. The network finetuning step uses training samples from train and val1 subsets. The val2 subset is used for evaluating components and the performance on test data is from the results submitted to the ImageNet challenge. Because the input for fast RCNN is an image from which both positive and negative samples are sampled, we discard images with no ground-truth boxes in the val1. Considering that lots of images in the train subset do not annotate all object instances, we reduce the number of images from this subset in the batch. For all networks, the learning rate and weight decay are fixed to 0.0005 during training, the batch size is 192. We use batch-based stochastic gradient descent to learn the network. The overhead time at inference due to gated connections is less than 40%.

B. Overall performance

1) ImageNet object detection dataset: We compare our framework with several other state-of-art approaches [4], [9], [30], [35], [42], [27]. The mean average precision for these approaches are shown in Table I. Our work is trained using the provided data of ImageNet. Compared with the published results and recent results in the provided data track on ImageNet 2015 challenge, our single model result performs better than the ResNet [27] by 4.5% in mAP for single-model result.

TABLE IV
OBJECT DETECTION MAP (%) ON MS-COCO FOR STATE-OF-TH-ART APPROACHES.

Method	Training Data	Network	mAP (%)	
			val	test-dev
FRCN [13]	train	VGG-16	–	35.9
Faster RCN [17]	train	VGG16	41.5	42.1
	train+val	VGG-16	–	42.7
Faster RCN [27]	train	ResNet-101	48.4	–
	train+val	ResNet-101	–	55.7
ION [10]	train	VGG16	–	44.7
ION-c [10]	train+val35k	VGG16	–	53.4
SSD [25]	train+val35k	VGG-16	–	46.5
GBD	train	ResNet-269	???	???
	train+val35k	ResNet-269	–	???
	train+val	ResNet-269	–	???

Table II shows the experimental results for UvA, GoogleNet, ResNet, which are best performing approaches in the ImageNet challenge 2013, 2014 and 2015 respectively. The top-10 approaches attending the challenge 2016 are also shown in Table II. Our approach has the similar mAP as Hikvision in single model and performs better for averaged model. Among the 200 categories, our submission wins 109 categories in detection accuracy.

2) PASCAL VOC2007 dataset: It contains 20 object categories. Following the most commonly used approach in [4], we finetune the Inception-v2 with the 07+12 trainval set and evaluate the performance on the test set. Our GBD-net obtains 77.2% mAP while the baseline Inception-v2+FRCN is only 73.1%.

3) Microsoft COCO object detection dataset: Table IV shows the experimental results for Fast R-CNN[13], Faster R-CNN [17] using VGG and ResNet, SSD [25], ION [10] and ION's COCO competition version, denoted by ION-c. Our GBD-Net performs better than these approaches. To investigate on the effectiveness of GBD-net, we use MCG [44] for region proposal and report both the overall AP and AP^{50} on the closed-test data. The baseline Inception-v2+FRCN implemented by us obtains 24.4% AP and 39.3% AP^{50}, which is comparable with Faster RCNN (24.2% AP) on COCO detection leadboard. With our proposal gated bi-directional structure, the network is improved by 2.6% AP and reaches 27.0% AP and 45.8% AP^{50}, which further shows the effectiveness of our GBD model.

C. Investigation on different settings in GBD-v1

1) Investigation on gate functions: Gate functions are introduced to control message passing for individual candidate

TABLE II
Object detection mAP (%) on ImageNet for the approaches attending the ImageNet challenge with single model (sgl) and averaged model (avg) when tested on the val2 data and Test data without using external data for training.

Year	2013	2014	2015	2016	2016	2016	2016	2016	2016	2016	2016	2016	2016
Team	UvA	GoogleNet	ResNet	VB	Faceall	MIL_UT	KAIST-SLSP	CIL	360+MCG	Trimps	NUIST	Hikvision	Ours
val2 (sgl)	-	38.8	60.5	-	49.3	-	-	-	-	-	-	65.1	65
val2 (avg)	-	44.5	63.6	-	52.3	-	-	-	-	-	-	67	68
Test (avg)	-	38	62.1	48.1	48.9	53.2	53.5	55.4	61.6	61.8	60.9	65.2	66.3
Test (sgl)	22.6	43.9	58.8	-	46.1	-	-	-	59.1	58.1	-	63.4	63.4

TABLE III
Detection mAP (%) for features with different padding values p for GBD-Net-v1 using Inception-v2 as the baseline. Craft-V1 is used for region proposal. GBD-Net-v1 is used when multiple resolutions are used. Different ps lead to different resolutions and contexts.

Padding value p	Single resolution				Multiple resolutions			
	-0.2	0.2	0.8	1.7	-0.2+0.2	0.2+1.7	-0.2+0.2+1.7	-0.2+0.2+0.8+1.7
mAP	46.3	46.3	46.0	45.2	47.4	47.0	48.0	48.9

boxes. With gate functions, the mAP is 48.9% when the GBD-Net places roi-pooling after the 6th inception module. Without gate functions, it is hard to train the network with message passing layers in our implementation. It is because nonlinearity increases significantly by message passing layers and gradients explode or vanish, just like it is hard to train RNN without LSTM (gating). In order to verify it, we tried different initializations. The network with message passing layers but without gate functions has 42.3% mAP if those message passing layers are randomly initialized. However, if those layers are initialized from a well-trained GBD-net, the network without gate functions reaches 48.2% mAP. These two results show the effectiveness of gate functions in making the model trainable.

2) Investigation on using different feature region sizes: The goal of our proposed gated bi-directional structure is to pass messages among features with different resolutions and contexts. In order to investigate the influence from different settings of resolutions and contexts, we conduct a series of experiments. In these experiments, features of a particular padding value p is added one by one. The experimental results for these settings are shown in Table III. When single padding value is used, it can be seen that simply enlarging the support region of CNN by increasing the padding value p from 0.2 to 1.7 does harm to detection performance because it loses resolution and is influenced by background clutter. On the other hand, integrating features with multiple resolutions and contexts using our GBD-Net substantially improves the detection performance as the number of resolutions/contexts increases. Therefore, with the GBD-Net, features with different resolutions and contexts help to validate the existence of each other in learning features and improve detection accuracy.

3) Investigation on combination with multi-region: This section investigates experimental results when combing our gated bi-directional structure with the multi-region approach. We adopt the simple straightforward method and average the detection scores of the two approaches. The baseline Inception-v2 model has mAP 46.3%. With our GBD-Net the mAP is 48.9%. The multi-region approach based on Inception-v2 has mAP 47.3%. The performance of combining our GBD-Net with mutli-region Inception-v2 is 51.2%, which has 2.3% mAP improvement compared with the GBD-Net and 3.9% mAP improvement compared with the multi-region Inception-v2. This experiment shows that the improvement brought by our GBD-Net is complementary to the multi-region approach in [11].

D. Investigation on GBD-v1 and GBD-v2

This section investigates the experimental results for the GBD-v1 in Section III-D and the GBD-v2 introduced in Section III-E. For the same Inception-v2 baseline, the GBD-v1 introduced in Section III improves the Inception-v2 by 2.6% in mAP. The GBD-v2 structure introduced in Section III-E improves mAP by 3.7%. Therefore, the GBD-v2 is better in improving the detection accuracy. Since GBD-v2 has better performance and is easier to train, we use the GBD-v2 as the default GBD structure in the following part of this paper if not specified. Since the components investigated in Section V-C are not different for GBD-v1 and GBD-v2, we directly adopt the settings found to be effective in GBD-v1 for GBD-v2. In GBD-v2, roi-pooling is placed at the module (4d) for Inception-v2 and the 234th layer for ResNet-269. Gate function is used. We use feature regions with padding values $p =$ -0.2, 0.2, 0.8, 1.7.

E. Investigation on other components in the detection framework

In the component-wise study, none of the techniques in Section IV-E is included if not specified. We adopt the left-right flip at the the training stage for data augmentation for all of the evaluated approaches but did not use flip at the testing stage if not specified.

1) Region proposal: We list the improvements on top of our early work Craft-v2 [18] in Table V. Random crop in training and multi-scale testing also help and they lead to 0.74% and 1.1% gain in recall, respectively. In multi-scale training, we want to keep the distribution of image size after log operation to be uniform. Therefore, in each iteration, we randomly select a scale number r in range of $[16, 512]$ and randomly select an object in a image with the length l. Then the resize scale is set to be l/r this image. This multi-scale training improves recall by 0.7%.

TABLE V

RECALL RATE ON IMAGENET VAL2 WITH DIFFERENT COMPONENTS IN PROPOSAL GENERATION.

Components.	baseline (Craft-v2) [18]	+Random Crop training	+Multi-scale testing	+Balancing positive and negative samples	+ensemble
Recall @ 150 proposals	92.37%	93.11%	94.21%	94.81%	
Recall @ 300 proposals					95.30%

TABLE VI

OBJECT DETECTION MAP (%) ON IMAGENET VAL2 FOR INCEPTION-V2 USING DIFFERENT PRETRAINING SCHEMES.

Pretraining scheme	Image	Object w/o roi	Object+roi
Region proposal	Craft-V2	Craft-V2	Craft-V2
mAP	49.1	48.4	49.9

In the testing stage, we densely resize the longer side of each image to $2800 \times 2^{(-9:0)}$ and found that it is necessary to achieve high enough recall for objects ranging in $[20, 50]$ pixels for longer side.

To balance the positive and negative samples, we implement a new multi-GPU implementation, where 50% of the GPUs only train positive samples while the other 50% GPUs only train the negative ones. Balancing the positive and negative samples to $1:1$ in training leads to 0.6% gain in recall.

We use 150 proposals for each image generated by the Craft framework. We combine two methods to get 300 proposals for each image and named it as Craft-v3. For the Craft-v3, the recall on ImageNet val2 is 95.30%, and the average recall [1] is 1.59.

Use the same Inception-v2 as the baseline on ImageNet val2, Craft-V1, V2 and V3 have mAP 46.3, 48.4, and 49.4 respectively. Compared to the Craft-v2 on ImageNet val2, the Craft-v3 leads to 1.9% gain in final detection AP for ResNet-269+GBD-Net.

2) Pretraining scheme: There are three pretraining schemes evaluated in the experimental results shown in Table VI. The Inception-v2 is used as the network for evaluation. The *image* in Table VI denotes the pretraining on the 1000-class image-centric classification task without using the box annotation of these objects. *Object w/o roi* denotes the pretraining on the 1000-class object-centric classification task using the box annotations without including the roi-pooling at the pretraining stage. *Object+roi* denotes the pretraining for the 1000-class object-centric classification task with the box annotations and with the roi-pooling included at the pretraining stage. Without using the roi-pooling at the pretraining stage, the object-centric pretraining performs worse than image-centric pretraining. The inclusion of roi-pooling at the pretraining stage improves the effectiveness of object centric pretraining for finetuning in object detection, with 1.5% increase in absolute mAP.

3) The GBD structure: We evaluate the GBD structure for different baseline models. Fig. VII shows the experimental results for different baseline networks with the GBD structure. The GBD structure introduced in Section III-E improves the Inception-v2 by 3.7% in mAP for Craft-V2 and by 4.2% in mAP for Craft-V3. With GBD and the better region proposal, Inception-v2+GBD with mAP 53.6% is close to ResNet-152 with mAP 54% in detection accuracy. The modified GBD structure improves the mAP by 2.5% and 2.2% for ResNet-152 and ResNet-269 respectively.

TABLE VII

OBJECT DETECTION MAP (%) ON IMAGENET VAL2 FOR DIFFERENT BASELINE NETWORKS WITH THE GBD-V2.THE + *new GBD* DENOTES THE USE OF THE MODIFIED GBD STRUCTURE IN FIG. 7 AND INTRODUCED IN SECTION III-E.

Baseline network	Inception-v2	Inception-v2	ResNet-152	ResNet-269
Region proposal	Craft-V2	Craft-V3	Craft-V2	Craft-V2
Pretrain	Object w/o roi	Object w/o roi	Image	Image
Without GBD	48.4	49.4	54	56.6
+ new GBD	52.1	53.6	56.5	58.8

It is mentioned in Section III-E that the magnitude of the messages from other contextual features influences the detection accuracy. Table VIII shows the experimental results for this influence. In the experiments, the Inception-v2 pretrained with bounding box label without roi-pooling is used as the baseline model. It can be seen that the scalar β has the best performance when it is 0.1. Setting β to be 1, i.e. not scaling messages, results in 1.6% mAP drop.

4) Baseline deep models: In this section, we evaluate the influence of baseline deep models for the detection accuracy on the ImageNet. Table IX shows the experimental results for different baseline network structures. All models evaluated are pretrained from ImageNet 1000-class training data without using the bounding box label. None of the model uses the stochastic depth [41] at the finetuning stage. If stochastic depth is included, it is only used at the pretraining stage. From the results in IX, it can be seen that ResNet-101 with identity mapping [40] and stochastic depth has 1.1% mAP improvement compared with the ResNet-101 without them. Because of time limit and the evidence in ResNet-101, we have used the stochastic depth and identity mapping for the ResNet-269 baseline model.

5) Model ensemble: For model ensemble, we have used six models and the averge of their scores are used as the result for model ensemble. As shown in Table X, these models vary in baseline model, pretraining scheme, use of GBD or not and region proposal for training the model. Note that the region proposal for training could be different, but they are tested using the same region proposal. Without context, the averaged model has mAP 66.9%. With global contextual scores, the model has mAP 68%.

6) Components improving performance: Table XI summarizes experimental results for the components that improve the performance. The baseline ResNet-269 has mAP 56.6%. With GBD-net, the mAP is 58.8%. Changing the region proposal from Craft-v2 to Craft-v3 improves the mAP to 60.7%. In the experimental results for the settings above, single-scale testing is used, in which the shorter side of the is constrained to be no greater than 600 and the longer is constrain to be no greater than 700 at the testing and training stage. The multi-scale testing introduced in Section IV-E1 provides 1.3% mAP improvement. Left-right flip provides 0.7 mAP gain. Bounding box voting leads to 1.3 % mAP gain. Changing the NMS threshold from 0.3 to 0.4 leads to 0.4 mAP gain. The use

[1] please refer more details of the proposal average recall to [45].

TABLE VIII
OBJECT DETECTION MAP (%) ON IMAGENET VAL2 FOR INCEPTION-V2 USING GBD STRUCTURE WITH DIFFERENT SCALE FACTOR β IN CONTROLLING THE MAGNITUDE OF MESSAGE.

Scale factor	$\beta = 1$	$\beta = 0.5$	$\beta = 0.2$	$\beta = 0.1$	β=0 (without GBD)
Pretrain	Object w/o roi	Object w/o roi	Object w/o roi	Object w/o roi	Object w/o roi
Region proposal	Craft-V3	Craft-V3	Craft-V3	Craft-V3	Craft-V3
mAP on val2	52	53.2	53.3	53.6	49.4

TABLE IX
OBJECT DETECTION MAP (%) ON IMAGENET VAL2 FOR DIFFERENT BASELINE DEEP MODELS. ALL MODELS ARE PRETRAINED FROM IMAGENET 1000-CLASS CLASSIFICATION DATA WITHOUT USING THE BOUNDING BOX LABEL. NONE OF THE APPROACHES INTRODUCED IN SECTION IV-E ARE USED. '+I' DENOTES THE USE OF IDENTITY MAPPING [40]. '+S' DENOTES THE USE OF STOCHASTIC DEPTH [41].

Net structure	Inception-v2	ResNet-101	ResNet-101+I+S	ResNet-152	ResNet-269+I+S	Inception-V5	PolyNet
	[30]	[27]	[40], [41]	[27]		[46]	[28]
Pretraining scheme	Image	Image	Image	Image	Image	Image	Image
Region proposal	Craft-V2	Craft-V2	Craft-V2	Craft-V2	Craft-V2	Craft-V2	Craft-V2
Mean AP	49.9	52.7	53.8	54	56.6	53.3	56.5

TABLE X
MODELS USED IN THE MODEL ENSEMBLE, GLOBAL CONTEXTUAL SCORES ARE NOT USED IN THE RESULTS FOR THESE MODELS. MODELS 1 AND 3 HAVE THE SAME NETWORK STRUCTURE BUT DIFFERENT INITIALIZATION. SIMILARLY FOR MODELS 2 AND 4.

Model denotation	1	2	3	4	5	6
Baseline model	ResNet-269	ResNet-269	ResNet-269	ResNet-269	PolyNet	ResNet-101
Use GBD	✓	✓	✓	✓		
Pretraining scheme	Object + roi	Image	Object + roi	Image	Image	Image
Region proposal for training	Craft-V3	Craft-V2	Craft-V3	Craft-V2	Craft-V3	Craft-V3
Averaged model	1	1+2	1+2+3	1+2+3+4	1+2+3+4+5	1+2+3+4+5+6
Mean AP (%)	63.5	64.8	65.5	66	66.8	66.9

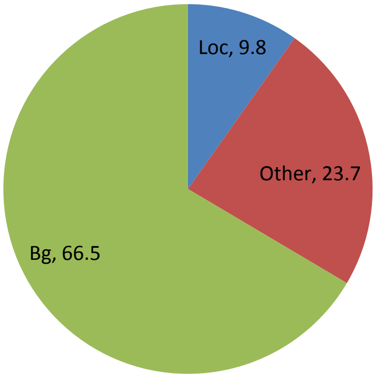


Fig. 9. Fraction of high-scored false positives on ImageNet Val2 that are due to poor localization (Loc), confusion with other objects (Other), or confusion with background or unlabeled objects (Bg)

of context provides 1.3% mAP improvement. The final single model result has 65% mAP on the val2 data. Ensemble of six models improves the mAP by 3% and the final result has 68% mAP.

F. Analysis of false positive types

Fig. 9 shows the fraction of false positives on ImageNet Val2 that are caused by confusion with background, poor localization and confusion with other objects. It can be seen that, the majority of false positives are from background, which is different from the results in [35] for Pascal VOC, where the majority of false positives are from poor localization. This is possibly from a better region proposal used in our approach.

VI. CONCLUSION

In this paper, we propose a gated bi-directional CNN (GBD-Net) for object detection. In this CNN, features of different resolutions and support regions pass messages to each other to validate their existence through the bi-directional structure. And the gate function is used for controlling the message passing rate among these features. Our GBD-Net is a general layer design which can be used for any network architecture and placed after any convolutional layer for utilizing the relationship among features of different resolutions and support regions. The effectiveness of the proposed approach is validated on three object detection datasets, ImageNet, Pascal VOC2007 and Microsoft COCO.

VII. ACKNOWLEDGMENT

This work is supported by SenseTime Group Limited and the General Research Fund sponsored by the Research Grants Council of Hong Kong (Project Nos. CUHK14206114, CUHK14205615, CUHK417011, and CUHK14207814).

REFERENCES

[1] Everingham, M., Gool, L.V., I.Williams, C.K., J.Winn, Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV **88**(2) (2010) 303–338

[2] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. IJCV (2015)

[3] 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: ECCV. (2014)

[4] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014)

[5] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324

[6] Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012)

[7] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)

TABLE XI
SUMMARY OF THE COMPONENTS THAT LEAD TO THE FINAL SUBMISSION.

ResNet-269	✓	✓	✓	✓	✓	✓	✓	✓	✓	ResNet-269
Craft-v2	✓	✓								Craft-v2
Craft-v3			✓	✓	✓	✓	✓	✓	✓	Craft-v3
Use GBD		✓	✓	✓	✓	✓	✓	✓	✓	Use GBD
Multi-scale testing				✓	✓	✓	✓	✓	✓	Multi-scale testing
Left-right flip					✓	✓	✓	✓	✓	Left-right flip
Box voting						✓	✓	✓	✓	Box voting
NMS threshold	0.3	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	NMS threshold
Context								✓	✓	Context
Model ensemble									✓	Model ensemble
Absolute mAP gain (%)	-	2.2	1.9	1.3	0.7	1.3	0.4	1.3	3	Absolute mAP gain
mAP (%)	56.6	58.8	60.7	62	62.7	63.3	63.7	65	68	mAP

[8] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[9] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. (2015)

[10] Bell, S., Zitnick, C.L., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: CVPR. (2016)

[11] Gidaris, S., Komodakis, N.: Object detection via a multi-region and semantic segmentation-aware cnn model. In: ICCV. (2015)

[12] Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. IEEE Trans. PAMI **30** (2013) 1915–1929

[13] Girshick, R.: Fast r-cnn. In: CVPR. (2015)

[14] Zeng, X., Ouyang, W., Yang, B., Yan, J., Wang, X.: Gated bi-directional cnn for object detection. In: ECCV. (2016)

[15] Smeulders, A., Gevers, T., Sebe, N., Snoek, C.: Segmentation as selective search for object recognition. In: ICCV. (2011)

[16] Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: ECCV. (2014)

[17] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. NIPS (2015)

[18] Yang, B., Yan, J., Lei, Z., Li, S.Z.: Craft objects from images. In: CVPR. (2016)

[19] Pont-Tuset, J., Van Gool, L.: Boosting object proposals: From pascal to coco. In: ICCV. (2015)

[20] Zhang, Y., Sohn, K., Villegas, R., Pan, G., Lee, H.: Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. In: CVPR. (2015)

[21] Gidaris, S., Komodakis, N.: Locnet: Improving localization accuracy for object detection. In: CVPR. (2016)

[22] Gidaris, S., Komodakis, N.: Attend refine repeat: Active box proposal generation via in-out localization. In: BMVC. (2016)

[23] Szegedy, C., Reed, S., Erhan, D., Anguelov, D.: Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441 (2014)

[24] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640 (2015)

[25] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.: Ssd: Single shot multibox detector. In: ECCV. (2016)

[26] Zagoruyko, S., Lerer, A., Lin, T.Y., Pinheiro, P.O., Gross, S., Chintala, S., Dollár, P.: A multipath network for object detection. arXiv preprint arXiv:1604.02135 (2016)

[27] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)

[28] Zhang, X., Li, Z., Loy, C.C., Lin, D.: Polynet: A pursuit of structural diversity in very deep networks. arXiv preprint arXiv:1611.05725 (2016)

[29] Chu, X., Ouyang, W., Li, H., Wang, X.: Structured feature learning for pose estimation. In: CVPR. (2016)

[30] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML (2015)

[31] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. arXiv preprint arXiv:1502.01852 (2015)

[32] Ouyang, W., Li, H., Zeng, X., Wang, X.: Learning deep representation with large-scale attributes. In: ICCV. (2015)

[33] Li, H., Ouyang, W., Wang, X.: Multi-bias non-linear activation in deep neural networks. In: ICML. (2016)

[34] Ouyang, W., Wang, X., Zhang, C., Yang, X.: Factors in finetuning deep model for object detection. In: CVPR. (2016)

[35] Ouyang, W., Wang, X., Zeng, X., Qiu, S., Luo, P., Tian, Y., Li, H., Yang, S., Wang, Z., Loy, C.C., et al.: Deepid-net: Deformable deep convolutional neural networks for object detection. In: CVPR. (2015)

[36] Girshick, R., Iandola, F., Darrell, T., Malik, J.: Deformable part models are convolutional neural networks. In: CVPR. (2015)

[37] Zeng, X., Ouyang, W., Wang, X.: Window-object relationship guided representation learning for generic object detections. arXiv preprint arXiv:1512.02736 (2015)

[38] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8) (1997) 1735–1780

[39] Bengio, Y., Glorot, X.: Understanding the difficulty of training deep feedforward neuralnetworks. In: AISTATS. (2010)

[40] He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: ECCV. (2016)

[41] Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.: Deep networks with stochastic depth. arXiv preprint arXiv:1603.09382 (2016)

[42] Yan, J., Yu, Y., Zhu, X., Lei, Z., Li, S.Z.: Object detection by labeling superpixels. In: CVPR. (2015)

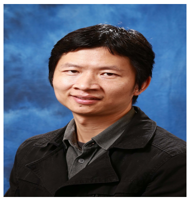
[43] He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: ECCV. (2014)

[44] Arbeláez, P., Pont-Tuset, J., Barron, J.T., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR. (2014)

[45] Hosang, J., Benenson, R., Dollár, P., Schiele, B.: What makes for effective detection proposals? IEEE Trans. PAMI **38**(4) (2016) 814–830

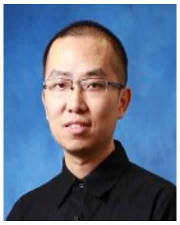
[46] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. arXiv preprint arXiv:1512.00567 (2015)

Xingyu Zeng received the B.S. degree in Electronic Engineering and Information Science from University of Science and Technology of China in 2011. He is currently a PHD student in the Department of Electronic Engineering at the Chinese University of Hong Kong. His research interests include computer vision and deep learning.



Wanli Ouyang received the PhD degree in the Department of Electronic Engineering, The Chinese University of Hong Kong, where he is now a Research Assistant Professor. His research interests include image processing, computer vision and pattern recognition.

Junjie Yan received his PhD degree in 2015 from National Laboratory of Pattern Recognition, Chinese Academy of Sciences. Since 2016, he is the principle scientist in SenseTime Group Limited and leads the research in object detection and tracking, and applications in video surveillance.



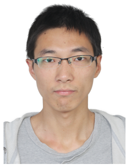
Hongsheng Li received the master's and doctorate degrees in computer science from Lehigh University, Pennsylvania, in 2010 and 2012, respectively. He is an associate professor in the School of Electronic Engineering at University of Electronic Science and Technology of China. His research interests include computer vision, medical image analysis, and machine learning.

Tong Xiao is a Ph.D. candidate in electronic engineering at The Chinese University of Hong Kong, advised by Prof. Xiaogang Wang. Tong received Bachelor of Engineering in computer science from Tsinghua University. His research interests include computer vision and deep learning, especially learning better feature representations for human identification. He also served as a reviewer of top-tier computer vision conferences and journals, including CVPR, ECCV, CVIU, and TCSVT.

Kun Wang received the B.S. degree in Software Engineering from Beijing Jiaotong University in 2014. He is currently a MPhil student in the Department of Electronic Engineering at the Chinese University of Hong Kong. His research interests include crowd analysis and object detection with deep learning.

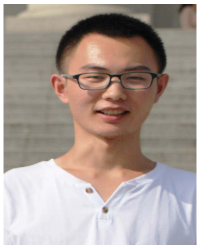
Yu Liu received the bachelor degree from School of Computer Science at Beihang University in 2016. He is currently a computer vision researcher in Sensetime Co. LTD. and will be a PhD student in the Department of Electronic Engineering at The Chinese University of Hong Kong in 2017. His research interests include computer vision and machine learning.

Yucong Zhou received the bachelor degree from School of Mathematics and Systems Science in 2015 and is pursuing the master degree with School of Computer Science at Beihang University. He is currently a computer vision intern at Sensetime Co. LTD. His research interests include computer vision and machine learning.



Bin Yang received the B.Eng. degree in Electronic and Information Engineering from China Agricultural University in 2014. From 2014 to 2015, he was with the Institute of Automation, Chinese Academy of Sciences, as a Research Assistant. He is currently pursuing the M.Sc. degree with the Computer Science Department, University of Toronto. His research interests are computer vision and machine learning.

Zhe Wang received the BEng in Optical Engineering from Zhejiang University, China, in 2012. He is currently working toward a PhD degree in the Department of Electronic Engineering, the Chinese University of Hong Kong. His research interest is focused on compressed sensing, magnetic resonance imaging, image segmentation and object detection.



Hui Zhou received the bachelor degree at university of science and electronic technology of china in 2015. He is currently a Research Assistant in the Department of Electronic Engineering at The Chinese University of Hong Kong. His research interests include computer vision and machine learning.

Xiaogang Wang received the PhD degree from the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology in 2009. He is currently an Assistant Professor in the Department of Electronic Engineering at The Chinese University of Hong Kong. His research interests include computer vision and machine learning.