

# ScarfNet: Multi-scale Features with Deeply Fused and Redistributed Semantics for Enhanced Object Detection

Jin Hyeok, Yoo

Hanyang University

Email: jhyoo@sphanyang.ac.kr

Dongsuk, Kum

Korea Advanced Institute of Science and Technology

Email: dskum@kaist.ac.kr

Jun Won, Choi

Hanyang University

Email: junwchoi@hanyang.ac.kr

**Abstract**—Convolutional neural networks (CNNs) have led us to achieve significant progress in object detection research. To detect objects of various sizes, object detectors often exploit the hierarchy of the multiscale feature maps called feature pyramids, which are readily obtained by the CNN architecture. However, the performance of these object detectors is limited because the bottom-level feature maps, which experience fewer convolutional layers, lack the semantic information needed to capture the characteristics of the small objects. To address such problems, various methods have been proposed to increase the depth for the bottom-level features used for object detection. While most approaches are based on the generation of additional features through the top-down pathway with lateral connections, our approach directly fuses multi-scale feature maps using bidirectional long short-term memory (biLSTM) in an effort to leverage the gating functions and parameter-sharing in generating deeply fused semantics. The resulting semantic information is redistributed to the individual pyramidal feature at each scale through the channel-wise attention model. We integrate our semantic combining and attentive redistribution feature network (ScarfNet) with the baseline object detectors, i.e., Faster R-CNN, single-shot multibox detector (SSD), and RetinaNet. Experimental results show that our method offers a significant performance gain over the baseline detectors and outperforms the competing multiscale fusion methods in the PASCAL VOC and COCO detection benchmarks.

## I. INTRODUCTION

Object detection refers to the task of deciding whether there are any instances of objects in the image and returns the estimate of the location and the category of the objects [1], [2]. Historically, object detection has been one of the most challenging computer vision problems. Recently, deep learning has led unprecedented advances in object detection techniques [2]. Convolutional neural networks (CNNs) can produce a hierarchy of abstract feature maps through a cascade of convolution operations followed by a nonlinear function. Using CNN as a backbone network, object detectors can effectively infer the location of the bounding box and the category of the instances based on the abstract feature maps. Various object detection network structures have been proposed to date. The CNN-based object detectors are roughly categorized into two groups: two-stage detectors and single-stage detectors. The two-stage detectors detect the objects using two separate subnetworks; 1) the region proposal network for finding the bounding boxes containing the object, and 2) the object classifier network for

identifying the class of the objects and refining the bounding boxes. The well-known two-stage detectors include R-CNN [3], Fast R-CNN [4], Faster R-CNN [5], and Mask R-CNN [6]. Single-stage detectors directly estimate the bounding boxes and the object classes from the feature maps in one shot and include single-shot multibox detector (SSD) [7], YOLO [8], YOLOv2 [9], and RetinaNet [10].

Recent advances in object detection are achieved by the CNN's capability to produce the abstract features containing strong semantic cues. The deeper the convolutional layers are, the higher the level of abstraction is for the resulting feature maps. As a result, the features produced at the end of the CNN pipeline (called *top-level features*) contain rich semantics but lack spatial resolution, whereas the features placed at the input layers (called *bottom-level features*) lack semantic information but have detailed spatial information. The hierarchy of such multiscale features constitutes so-called *feature pyramids*, which are used to detect the objects of different scales in many object detectors (e.g., SSD [7], MSCNN [11], and RetinaNet [10]). The structure designed to use such feature pyramids for object detection is described in Fig. 1 (a). Note that the attributes of the large objects tend to be captured on the top-level features of small size while those of the small objects are well represented by the shallow bottom-level features of large size.

One limitation of the feature pyramid method is the disparity of the semantic information between the multiscale feature maps used for object detection. The bottom-level features are not deep enough to exhibit high-level semantics underlying in the objects and their surroundings. This results in the accuracy loss in detecting the small objects. In order to address this problem, several approaches, which have attempted to reduce the semantic gap between the pyramidal features in different scales, have been proposed. One notable direction is to provide the contextual information to the bottom-level features by generating the highly semantic features in the top-down pathway with latent connections. As illustrated in Fig. 1 (c), based on the top-level pyramidal feature obtained from the bottom-up network, the additional features are subsequently generated with increased depth and resolution. In order to avoid losing the spatial information, lateral connections are used to take the bottom-level features and combine them

with the high-level semantic features. Various object detectors including DSSD [12], FPN [13], StairNet [14], Libra R-CNN [15], HR-FPN [16], and NAS-FPN [17] follow this principle, and significant improvement has been reported in terms of detection accuracy.

Our work is motivated by the observation that the capacity of the current architectures for generating top-down features might not be large enough to generate strong semantics for all scales. Thus, we propose a new architecture, which deeply fuses the multiscale features for enhanced object detection. The proposed feature pyramid method, referred to as *semantic combining and attentive redistribution feature network* (ScarfNet), combines the multiscale feature maps using the recurrent neural networks and then redistributes the fused semantics to each level, generating the new multiscale feature maps. The structure of ScarfNet is depicted in Fig. 2 (d). First, the bidirectional long short-term memory (biLSTM) model [18] is used to combine the multiscale pyramidal features. Although biLSTM is widely used to extract the temporal features from the sequential data, it can effectively combine the semantics in multiscale features. Our conjecture is that compared to the convolutive fusion methods, biLSTM requires significantly reduced number of weights due to parameter sharing, and the only relevant semantic information is selectively aggregated through the gating function of biLSTM. The fused feature maps are distributed through the channel-wise attention model, generating highly semantic features tailored for each pyramid scale. The final multiscale feature maps are used for object detection. Note that our framework can be readily applied to various feature pyramid-based CNN architectures, which require strong semantic information.

In the experiments, we integrate ScarfNet with the baseline detectors including Faster R-CNN [5], SSD [7], and RetinaNet [10]. The evaluation conducted over PASCAL VOC [19] and MS COCO [20] datasets shows that our method offers significant improvement over the baseline detectors as well as other competitive detectors in terms of detection accuracy. Our code will be made publicly available. The contributions of our paper are summarized as follows.

- We introduce a new deep architecture for closing the semantic gaps between the multiscale feature maps. The proposed ScarfNet generates new multiscale feature maps with deeply fused and redistributed semantics by using the combination of biLSTM and the channel-wise attention model.
- For the first time in the literature, the biLSTM is used to combine the multiscale features to incorporate strong semantics for feature pyramids. The biLSTM model can produce deeply fused semantic information using the recurrent connection over different pyramid scales. Furthermore, ScarfNet benefits from the selective information gating mechanism inherent in the biLSTM model. Due to parameter sharing, the overhead due to ScarfNet is small. In addition, ScarfNet is easy to train and is also end-to-end trainable.

## II. RELATED WORK

In this section, we review the basic object detectors and several existing feature pyramid methods used to decrease the semantic gap between the scales.

### A. CNN-based Object Detectors

Recently, CNN has brought an order of magnitude performance improvement in object detection. Various CNN-based object detectors have been proposed to date. The current object detectors can be categorized into two groups: two-stage detectors and single-stage detectors. Two-stage detectors detect the objects in two steps; finding the region proposals based on the objectness of the regions and conducting the classification and bound regression for detected region proposals. R-CNN [3] is the first CNN-based detector where the traditional selective search is employed to find the region proposals, and CNN is applied to the image patch in each region proposal. Fast R-CNN [4] and Faster R-CNN [5] reduced the computation time of R-CNN by employing the region of interest (ROI) pooling for using full-image feature maps and replacing the selective search with the region proposal network (RPN). Single-stage detectors directly perform classification and box regression based on the feature maps. These detectors compute the confidence score on the object category and the regression results for the candidate boxes while sweeping the feature maps spatially. The well-known single-stage detectors include SSD [7], YOLO [8], and YOLOv2 [9]. Recently, RetinaNet [10] has achieved the-state-of-the-art performance using ResNet [21] as a backbone and various latest training tricks. Refer to [2] for the comprehensive review of contemporary object detectors.

### B. Object Detectors Using Multiscale Features

Several object detectors including SSD [7] and RetinaNet [10] rely on hierarchical feature pyramids to detect the objects of various sizes (Fig. 1 (a)). One problem with using multiscale features directly produced by CNNs is the gap of the semantic information between them caused by the different depths of the layers passed by the input. Due to the relatively low level of abstraction for bottom-level features, detection accuracy for small objects is often limited. Fig. 1 (b), (c), and (d) describe the strategies proposed to overcome this problem. Fig. 1 (b) depicts the strategy of combining the multiscale features into the single high-resolution feature map with strong semantics. HyperNet [22] and ION [23] improved the performance of RPN by aggregating the hierarchical features with the appropriate resizing of the feature maps. Fig. 1 (c) shows the strategy of generating highly semantic features through the top-down pathway with lateral connections. Note that the semantic information is generated through the top-down connections while the detailed spatial information is provided through lateral connections. Several detectors based on this structure include DSSD [12], StairNet [14], TDM [24], FPN [13], and RefineDet [25]. DSSD [12], StairNet [14], and Libra R-CNN [15] use the deconvolutional layer-based top-down connections. TDM [24] employs the top-down structure specified for the RPN of the Faster R-CNN [5]. FPN [13]

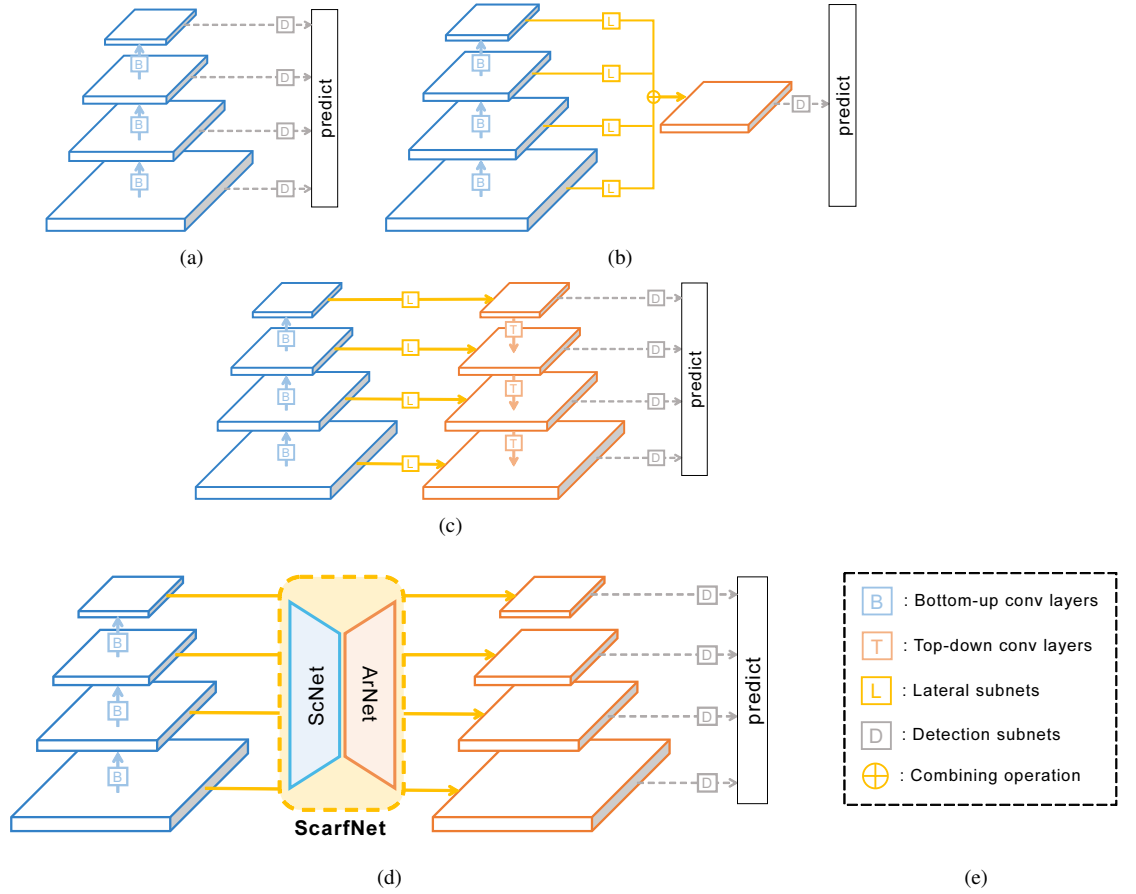


Fig. 1. **Structure of several feature pyramid methods:** In (a), feature pyramid obtained from convolutional layers is used in baseline detectors (e.g., SSD [7]). In (b), multiscale features are fused and converted into the single semantic feature map with the highest resolution. (c) shows the structure generating additional features in unidirectional way through the top-down structure with lateral connections. (d) shows the structure of the proposed ScarfNet, where the multiscale features are fused in a bidirectional fashion and the learned semantics are propagated back to each scale.

uses the simplified structure using 2x up-sampling and 1x1 convolution for top-down and lateral connections, respectively. RefineDet [25] employs two-step cascade regression for top-down connections. NAS-FPN [17] uses neural architecture search (NAS) to further optimize the process of designing FPN. HR-FPN [16] demonstrates the effectiveness of high resolution and multi-level features learned by the modified backbone networks.

### III. PROPOSED OBJECT DETECTOR

In this section, we introduce the details of the proposed ScarfNet architecture.

#### A. Existing Feature Pyramid Methods

The feature pyramid-based object detectors performs detection based on the  $k(> 1)$  feature maps across different pyramid levels to detect the various sizes of objects. As shown in Fig. 1 (a), the baseline detectors use the feature map  $X_l$  at the  $l$ th pyramidal level

$$X_l = B_l(X_{l-1}) \quad (1)$$

$$\text{Detection Outputs} = D_l(X_l), \quad (2)$$

where  $l = n - k + 1, \dots, n$ . Note that  $X_{1:n-k}(=[X_1, X_2, \dots, X_{n-k}])$  are the feature maps produced by the backbone network, and  $X_{n-k+1:n}$  are the bottom-up features from the subsequent convolutional layers.  $B_l(\cdot)$  denotes the operation performed by the  $l$ th convolutional layer, and  $D_l(\cdot)$  denotes the detection subnetwork that often applies a single 3x3 convolutional layer to produce the output of classification and box regression. Due to the different depths from the input to each pyramidal feature, shallow bottom-level features suffer from the lack of semantic information.

In order to reduce the semantic gap between different pyramid levels, several works proposed the top-down structure using lateral connections as illustrated in Fig. 1 (c). This structure propagates the high-level semantics from the top layers to the bottom layers with increased resolution while keeping the spatial high resolution through lateral connections. The  $l$ th feature map  $X'_l$  generated using this method is expressed as

$$X'_l = L_l(X_l) \oplus T_l(X'_{l+1}) \quad (3)$$

$$\text{Detection Outputs} = D_l(X'_l) \quad (4)$$

where  $l = n - k + 1, \dots, n$ . Note that  $L_l(\cdot)$  is the operation

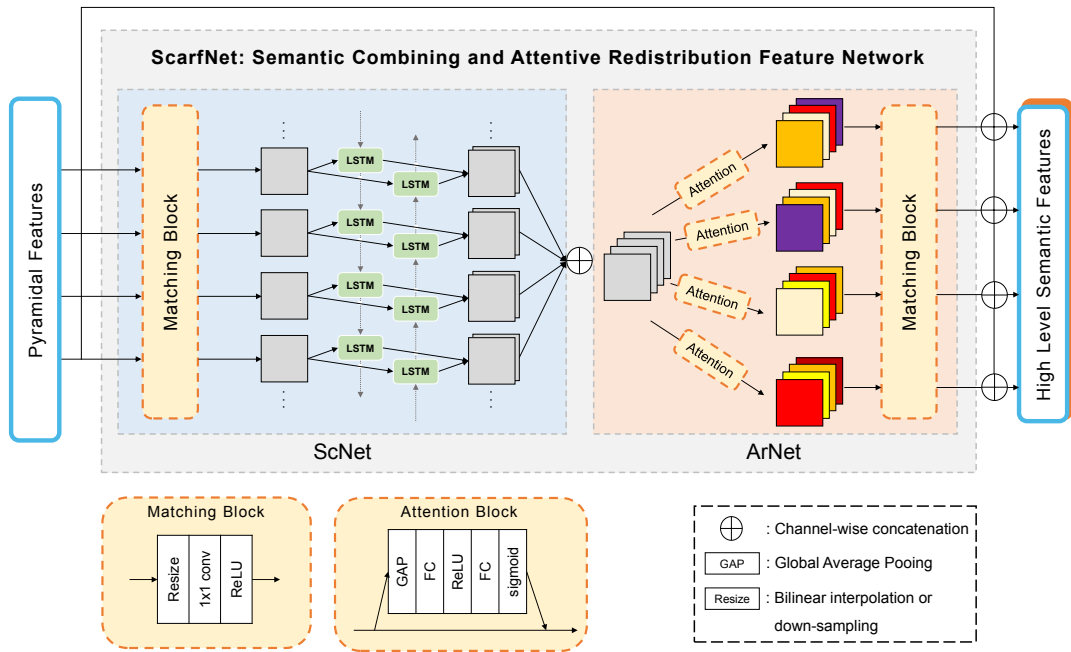


Fig. 2. **The overall architecture of the proposed ScarfNet:** ScarfNet consists of two modules: ScNet and ArNet. ScNet aggregates the pyramidal features obtained from the bottom-up CNN pipeline. Then, ArNet distributes the fused semantics to each pyramid level. The final high-level semantic features are generated by channel-wise concatenation between the output of ScarfNet and the original pyramidal features. The detailed structures of the matching block and the attention block are depicted in the yellow boxes.

for the  $l$ th lateral connection, and  $T_l(\cdot)$  is the operation for the  $l$ th top-down connection. The operator  $\oplus$  represents the operation of combining two feature maps, e.g., channel-wise concatenation and addition. Different methods (e.g., DSSD [12], StairNet [14], TDM [24], FPN [13], and RefineDet [25]) employ the slightly different structures for  $L_l(\cdot)$  and  $T_l(\cdot)$ . While these methods promote the abstraction level for pyramidal features, they still have some limitations. Since the top-down connection propagates the semantic information in a unidirectional way, the semantics are not evenly distributed among all pyramid levels. As a result, the semantic gap between the pyramidal features still remains. Next, such unilateral processing of the features has limited capacity to produce rich contextual information for increasing the semantic levels in all scales. In order to address these problems, we developed a new architecture that uses biLSTM to generate the deeply fused semantics through bi-lateral connections between all pyramid scales. In the following subsections, we present the details of our design.

### B. ScarfNet: Overall Architecture

ScarfNet attempts to resolve the discrepancy of the semantic information in two steps: 1) combining the scattered semantic information using biLSTM and 2) redistributing the fused semantics back to each pyramid level using the channel-wise attention model. The overall architecture of ScarfNet is depicted in Fig. 2. Taking the  $k$  pyramidal features  $X_{n-k+1:n}$

as input, ScarfNet produces the new  $l$ th pyramidal feature map  $X'_l$  as

$$X'_l = \text{ScarfNet}_l(X_{n-k+1:n}) \quad (5)$$

$$= X_l \oplus \text{ArNet}_l(\text{ScNet}(X_{n-k+1:n})) \quad (6)$$

$$\text{Detection Outputs} = D_l(X'_l) \quad (7)$$

where  $l = n - k + 1, \dots, n$ . As in (6), ScarfNet consists of two subnetworks: semantic combining network (ScNet) and attentive redistribution network (ArNet). First, ScNet merges the pyramidal features  $X_{n-k+1:n}$  through biLSTM and produces the output features with the fused semantics. Second, ArNet collects the output features from biLSTM and applies the channel-wise attention model to produce highly semantic multiscale features, which are concatenated to the original pyramidal features. Finally, the resulting feature maps are individually processed by the detection subnetwork  $D_l(\cdot)$  to produce the results for object detection.

### C. Semantic Combining Network (ScNet)

The feature maps  $X_{n-k+1:n}^f$  produced by ScNet are obtained as follows:

$$X_{n-k+1:n}^f = \text{ScNet}(X_{n-k+1:n}), \quad (8)$$

where  $X_l^f$  is the output feature map for the  $l$ th layer. Fig. 3 depicts the detailed structure of ScNet. ScNet uniformly fuses

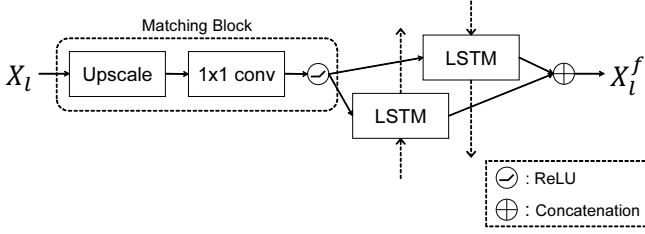


Fig. 3. **The structure of ScNet:** The matching block and biLSTM are applied to generate the fused feature map  $X_l^f$ . Note that the matching block applies bilinear interpolation and 1x1 convolution to make the spatial and channel dimensions equal for the inputs to biLSTM.

the semantics scattered in different pyramid levels using biLSTM. The biLSTM model can selectively fuse the contextual information in multiscale features through the gating function. As shown in Fig. 3, ScNet consists of the matching block and the biLSTM block. The matching block first resizes the pyramidal features  $X_{n-k+1:n}$  such that they have the same size as the largest pyramidal feature. Then, it adjusts the channel dimension of the input using the 1x1 convolutional layer. As a result, the matching block produces the feature maps of the same spatial and channel dimensions for biLSTM. Note that the resizing operation is performed by bilinear interpolation. The biLSTM model used in SCNet follows the structure of [26], which has significantly saved the computation time by using convolutional layers for input connection and computing the gating parameters based on the results of global average pooling. The operations performed by biLSTM in [26] are summarized as follows:

$$\bar{X}_l = \text{GlobalAveragePooling}(X_l) \quad (9)$$

$$\bar{X}_{l-1}^f = \text{GlobalAveragePooling}(X_{l-1}^f) \quad (10)$$

$$i_l = \sigma \left( W_{xi} \bar{X}_l + W_{xf_i} \bar{X}_{l-1}^f + b_i \right) \quad (11)$$

$$f_l = \sigma \left( W_{xf} \bar{X}_l + W_{xf_f} \bar{X}_{l-1}^f + b_f \right) \quad (12)$$

$$o_l = \sigma \left( W_{xo} \bar{X}_l + W_{xf_o} \bar{X}_{l-1}^f + b_o \right) \quad (13)$$

$$G_l = \tanh \left( W_{xc} * X_l + W_{xf_c} * X_{l-1}^f + b_c \right) \quad (14)$$

$$C_l = X_l \circ C_{l-1} + i_l \circ G_l \quad (15)$$

$$X_l^f = o_l \circ \tanh(C_l), \quad (16)$$

where  $\circ$  denotes the Hadamard product. The state update of biLSTM is conducted in both forward and backward directions. Note that we only provide the forward update, and the equations are similar for the backward update.

#### D. Attentive Redistribution Network (ArNet)

The ArNet aims to produce the high-level semantic feature map, which is concatenated with the original pyramidal feature map  $X_l$  as follows:

$$X_l' = X_l \oplus \text{ArNet}_l(X_{n-k+1:n}^f), \quad (17)$$

where operator  $\oplus$  denotes the channel-wise concatenation. The detailed structure of ArNet is depicted in Fig. 4. ArNet

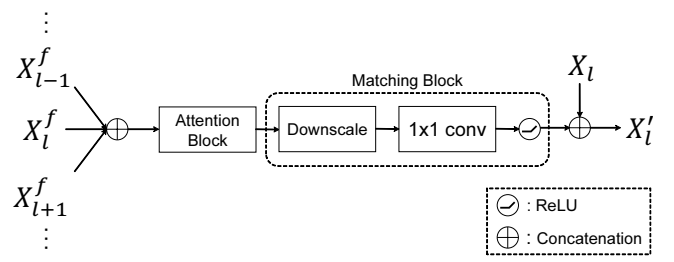


Fig. 4. **The structure of ArNet:** ArNet concatenates the fused feature maps  $X_{n-k+1:n}^f$  and applies the channel-wise attention. Then, the spatial and channel dimensions of the resulting feature maps are adjusted by the matching block.

concatenates the outputs  $X_{n-k+1:n}^f$  of biLSTM and apply the channel-wise attention to them. The attention weights are obtained by constructing the 1x1 vector using global average pooling [28] and passing it through two fully connected layers followed by the sigmoid function. Note that this channel-wise attention model allows for selective propagation of the semantics to each pyramid level. Once the attention weights are applied, the matching block down-samples the resulting feature maps to the original size of the pyramidal features and applies 1x1 convolution to match the channel dimensions with those of the original pyramidal features. Finally, the output of the matching block is concatenated with the original feature  $X_l$  to produce the highly semantic feature  $X_l'$ .

## IV. EXPERIMENTS

We evaluated the performance of the proposed ScarfNet model by comparing our detector with other multiscale detection methods and conducting an extensive performance analysis to understand the behavior of our architecture.

#### A. Experimental Setup

ScarfNet was applied to the baseline object detectors including Faster R-CNN [5], SSD [7], and RetinaNet [10]. In the case of Faster R-CNN and RetinaNet, we replaced the original FPN part with the feature generation by ScarfNet. We compared our method with the baselines, Faster R-CNN [5], SSD [7], and RetinaNet [10], as well as the other competitive algorithms including ION [23], R-FCN [27], DSSD [12], and StairNet [14]. We measured the mean average precision (mAP) in % on three widely used datasets for object detection benchmark: PASCAL VOC 2007, PASCAL VOC 2012 [19], and MS COCO [20].

#### B. Network Configuration

The advantage of ScarfNet is that there are not many hyperparameters to be determined. Note that the spatial dimensions of the feature maps are readily determined based on those of the baseline detectors. The channel dimensions of the intermediate feature maps are fixed over the pipeline between two matching blocks in ScNet and ArNet. Thus, we only need to determine this channel dimension. According to our empirical results, we set the channel dimension to 256.

Method	Backbone	Input size	mAP (%)	
			VOC 2007	VOC 2012
StairNet [14]	VGG-16	300 × 300	78.8	76.4
Faster R-CNN [5]	VGG-16	~ 1000 × 600	73.2	70.4
ION [23]	VGG-16	~ 1000 × 600	76.5	76.4
SSD300* [7]	VGG-16	300 × 300	77.5	75.8
<b>Scarf SSD300 (ours)</b>	VGG-16	300 × 300	79.4	77.2
SSD512* [7]	VGG-16	512 × 512	79.8	78.5
<b>Scarf SSD512 (ours)</b>	VGG-16	512 × 512	<b>81.6</b>	<b>79.8</b>
SSD321 [12]	ResNet-101	321 × 321	77.1	75.4
SSD513 [12]	ResNet-101	513 × 513	80.6	79.4
DSSD321 [12]	ResNet-101	321 × 321	78.6	76.3
DSSD513 [12]	ResNet-101	513 × 513	81.5	80.0
R-FCN [27]	ResNet-101	~ 1000 × 600	80.5	77.6
Faster R-CNN <sup>†</sup> [5]	ResNet-101	~ 833 × 500	81.1	-
<b>Scarf Faster R-CNN (ours)</b>	ResNet-101	~ 833 × 500	82.3	-
RetinaNet500 <sup>†</sup> [10]	ResNet-101	~ 833 × 500	83.0	-
<b>Scarf RetinaNet500 (ours)</b>	ResNet-101	~ 833 × 500	<b>83.5</b>	-

TABLE I

**PASCAL VOC 07/12 DETECTION RESULTS:** THE DETECTION RESULTS FOR VOC 2007 ARE EVALUATED ON *VOC 2007 test set* AFTER TRAINED ON *VOC 2007 trainval* AND *VOC 2012 trainval*. THOSE FOR VOC 2012 ARE EVALUATED ON *VOC 2012 test set* WHEN TRAINED ON *VOC 2007 test*, *VOC2007 trainval*, AND *VOC 2012 trainval* SETS.

### C. Performance Evaluation: PASCAL VOC Results

1) *Training on PASCAL VOC 2007 Dataset:* The object detectors under consideration were trained with the *VOC 2007 trainval* and the *VOC 2012 trainval* sets and evaluated with the *VOC 2007 test* set. When ScarfNet was combined with the SSD baseline, we trained our model over 120k iterations (~240 epochs). We used a learning rate of  $10^{-3}$  for the first 80k iterations,  $10^{-4}$  for the next 20k iterations, and  $10^{-5}$  for the last 20k iterations. The mini-batch size was set to 32, the momentum for the stochastic gradient descent (SGD) update was set to 0.9, and the weight decay was set to 0.0005. When our method was combined with the RetinaNet baseline, we used a learning rate of  $5 \times 10^{-3}$  for the first 60k iterations,  $5 \times 10^{-4}$  for the next 20k iterations, and  $5 \times 10^{-5}$  for the last 10k iterations. Other parameters were equally set except for the weight decay of 0.0001.

2) *Training on PASCAL VOC 2012 Dataset:* The object detectors were trained with the *VOC 2007 trainval*, the *VOC 2007 test* and the *VOC 2012 trainval* sets and evaluated with the *VOC 2012 test* set. When our model was applied to the SSD baseline, a total of 200k iterations were run with the same training parameters as in the VOC 2007 case. Note that we used a learning rate of  $10^{-3}$  for the first 120k iterations,  $10^{-4}$  for the next 40k iterations, and  $10^{-5}$  for the rest.

3) *Performance Comparison:* Table I shows the mAP performance of the object detectors under comparison evaluated on the PASCAL VOC 2007 and 2012 test sets. For both PASCAL 2007 and 2012 cases, the semantic features generated by ScarfNet offer a significant performance gain over the baseline detectors. In the case of PASCAL VOC 2007, the proposed method achieves 1.9%, 1.8%, and 1.2% mAP gains over the SSD300, SSD512, and Faster R-CNN baselines, respectively.

The proposed method also outperforms the RetinaNet baseline by 0.5%. Since the RetinaNet baseline employs the top-down structure based on FPN [13], we conclude that the features generated by our method are superior to those generated by FPN. Our object detector also achieves better performance than the other competing algorithms including, StairNet [14], DSSD [12], ION [23], R-FCN [27]. Although the detection accuracy with the PASCAL VOC 2012 dataset is slightly degraded compared to that with PASCAL VOC 2017, the tendency of the detection results observed for PASCAL VOC 2007 remains the same. Note that the proposed detector maintains the performance gain of 1.4% and 1.3% mAP over the SSD300 and SSD500 baselines, respectively.

### D. Performance Evaluation: COCO Results

1) *Training:* The object detectors under comparison were trained with the *MS COCO trainval35k* split [23] (union of 80k images from the training set and a random 35k subset of images from 40k image val split). The evaluation was performed using the *MS COCO test-dev*. To train the proposed structure based on RetinaNet [10], we used the learning rate of  $10^{-2}$  for the first 60k iterations,  $10^{-3}$  for the next 20k iterations, and  $10^{-5}$  for the last 20k iterations. The mini-batch size was set to 16, the momentum was set to 0.9, and the weight decay was set to 0.0001.

2) *Performance comparison:* Table II provides the detection accuracy of the algorithms tested on the MS COCO dataset. The experiment was conducted on various baseline detectors and feature pyramid modules. The proposed Scarf SSD513 and Scarf RetinaNet achieve the significant performance gain over the baselines. Our method beats the Faster R-CNN baseline [5] by 0.9% AP. Note also that the Scarf

Method	Network	Backbone	Module	Input size	fps	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
<i>two-stage</i>	Faster R-CNN* [5]	ResNeXt-101	FPN	$\sim 833 \times 500$	15.3	37.6	59.1	40.7	19.2	41.8	52.3
		ResNeXt-101	FPN	$\sim 1333 \times 800$	10.3	41.9	63.9	45.9	25.0	45.3	52.3
	Scarf Faster R-CNN (ours)	ResNeXt-101	SCARF	$\sim 833 \times 500$	13.8	38.5	59.9	41.5	19.1	42.9	<b>54.1</b>
		ResNeXt-101	SCARF	$\sim 1333 \times 800$	8.9	<b>42.8</b>	<b>64.3</b>	<b>47.1</b>	<b>26.0</b>	<b>45.7</b>	52.9
<i>one-stage</i>	SSD513 [12]	ResNet-101	-	$513 \times 513$	12.5	31.2	50.4	33.3	10.2	34.5	49.8
	DSSD513 [12]	ResNet-101	DSSD	$513 \times 513$	10.0	33.2	53.3	35.2	13.0	35.4	51.1
	Scarf SSD513 (ours)	ResNet-101	SCARF	$513 \times 513$	11.5	<b>34.5</b>	<b>54.1</b>	<b>36.3</b>	<b>15.1</b>	<b>36.1</b>	<b>51.6</b>
	RetinaNet [10]	ResNet-101	FPN	$\sim 833 \times 500$	15.4	34.4	53.1	36.8	14.7	38.5	49.1
		ResNeXt-101	FPN	$\sim 1333 \times 800$	9.3	40.8	61.1	44.1	24.1	44.2	51.2
	Scarf RetinaNet (ours)	ResNet-101	SCARF	$\sim 833 \times 500$	13.6	35.1	53.8	37.7	15.8	38.7	49.0
		ResNeXt-101	SCARF	$\sim 1333 \times 800$	8.4	<b>41.6</b>	<b>62.0</b>	<b>44.6</b>	<b>24.5</b>	<b>45.5</b>	<b>52.3</b>

TABLE II

**DETECTION RESULTS ON MS COCO test-dev DATASET:** THE SYMBOL “\*” INDICATES OUR RE-IMPLEMENTED RESULTS. THE EXPRESSION “ $\sim x \times y$ ” MEANS RE-SCALING OF THE INPUT IMAGE INTRODUCED IN THE ORIGINAL RETINANET PAPER.

	Method	mAP
Ablation study	Baseline (SSD)	77.5
	biLSTM	79.1
	biLSTM + channel-wise attention	<b>79.4</b>
Other fusion strategy (used with channel-wise attention)	1x1 conv.-based fusion	78.9
	uniLSTM	78.7
	Top-down structure with lateral connections	78.6

TABLE III

RESULTS OF ABLATION STUDY ON VOC 2007 test DATASET.

		Semantic feature generation strategy	
		Addition	Concat.
Channel dimension	64	78.3	78.8
	128	78.6	79.1
	256	79.1	<b>79.4</b>
	512	79.5	79.2
	1024	79.4	79.2

TABLE IV

MAP (%) PERFORMANCE FOR VARIOUS COMBINATIONS OF CHANNEL DIMENSION AND SEMANTIC FEATURE GENERATION STRATEGY WHEN EVALUATED ON VOC 2007 test set

SSD513 achieves 1.3% performance gain over DSSD513 and the Scarf RetinaNet offers the 0.8% performance gain over the RetinaNet baseline [10].

### E. Performance Analysis: Ablation Study

1) *Benefits of biLSTM:* It is worth investigating the effectiveness of biLSTM and channel-wise attention for fusing the multi-scale features. Table III shows how the performance of our method improves as we add bi-LSTM and channel-wise attention to the baseline one by one. We see that the biLSTM offers the 1.6% AP gain over the baseline and combination of biLSTM and channel-wise attention adds 1.9% AP gain. Table III also compares the different fusion strategies including the 1x1 convolutional layer, the top-down structure, and the uni-directional LSTM. Our biLSTM achieves better performance than the others; thus, parameter sharing, gating units, and bilateral processing of biLSTM effectively control high-level information to reduce the subtle semantic gap between the hierarchical features.

2) *Network Parameter Search:* We need to determine the channel dimension of the intermediate feature maps. We should also determine whether the element-wise addition or channel-wise concatenation is better in combining the output of ScarfNet with the original feature pyramid. Table IV shows the evaluation of the performance of our detector for various combinations of channel dimensions (64, 128, 256, 512 versus 1024) and feature combining strategies (element-wise addition versus channel-wise concatenation). According to Table IV,

the combination of 512 channel dimensions with element-wise addition leads to the best detection accuracy. However, using 512 channels significantly increases the computational complexity of the entire network; thus, we chose 256 channel dimensions with channel-wise concatenation.

## V. CONCLUSIONS

In this study, we developed a deep architecture that generates multiscale features with strong semantics to reliably detect the objects in various sizes. Our ScarfNet method transforms the pyramidal features produced by the baseline detector into evenly abstract features. ScarfNet fuses the pyramidal features using biLSTM and distributes the semantics back to each multiscale feature. We verified through experiments conducted with PASCAL VOC and MS COCO datasets that the proposed ScarfNet method significantly increases the detection performance over the baseline detectors. Our object detector achieves the state-of-the-art performance on the PASCAL VOC and COCO benchmarks.

## ACKNOWLEDGEMENT

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-01373, Artificial Intelligence Graduate School Program (Hanyang University)) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1A2C2012146).

## REFERENCES

- [1] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," 1998, pp. 555–562.
- [2] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *arXiv preprint arXiv:1809.02165*, 2018.
- [3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [4] R. Girshick, "Fast r-cnn," *IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015.
- [5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems (NIPS)*, pp. 91–99, 2015.
- [6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2017, pp. 2961–2969.
- [7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *European Conference on Computer Vision (ECCV)*, pp. 21–37, 2016.
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.
- [9] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, 2017.
- [10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017.
- [11] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," *European Conference on Computer Vision (ECCV)*, pp. 354–370, 2016.
- [12] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.
- [13] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [14] S. Woo, S. Hwang, and I. S. Kweon, "Stairnet: Top-down semantic aggregation for accurate one shot detection," *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1093–1102, 2018.
- [15] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-cnn: Towards balanced learning for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 821–830.
- [16] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, "High-resolution representations for labeling pixels and regions," *arXiv preprint arXiv:1904.04514*, 2019.
- [17] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Nas-fpn: Learning scalable feature pyramid architecture for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 7036–7045.
- [18] L. Zhang, G. Zhu, L. Mei, P. Shen, S. A. A. Shah, and M. Bennamoun, "Attention in convolutional lstm for gesture recognition," *Advances in Neural Information Processing Systems (NIPS)*, pp. 1957–1966, 2018.
- [19] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IEEE International Conference on Computer Vision (ICCV)*, vol. 88, no. 2, pp. 303–338, 2010.
- [20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," *European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [22] T. Kong, A. Yao, Y. Chen, and F. Sun, "Hypernet: Towards accurate region proposal generation and joint object detection," *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 845–853, 2016.
- [23] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2874–2883, 2016.
- [24] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta, "Beyond skip connections: Top-down modulation for object detection," *arXiv preprint arXiv:1612.06851*, 2016.
- [25] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4203–4212, 2018.
- [26] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in Neural Information Processing Systems (NIPS)*, pp. 802–810, 2015.
- [27] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," *Advances in Neural Information Processing Systems (NIPS)*, pp. 379–387, 2016.
- [28] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," 2018, pp. 7132–7141.