2018 IEEE Winter Conference on Applications of Computer Vision

StairNet: Top-Down Semantic Aggregation for Accurate One Shot Detection

Sanghyun Woo
KAIST
shwoo93@kaist.ac.kr

Soonmin Hwang
KAIST
jjang9hsm@kaist.ac.kr

In So Kweon
KAIST
iskweon@kaist.ac.kr

Abstract

One-stage object detectors such as SSD or YOLO already have shown promising accuracy with small memory footprint and fast speed. However, it is widely recognized that one-stage detectors have difficulty in detecting small objects while they are competitive with two-stage methods on large objects. In this paper, we investigate how to alleviate this problem starting from the SSD framework. Due to their pyramidal design, the lower layer that is responsible for small objects lacks strong semantics(e.g contextual information). We address this problem by introducing a feature combining module that spreads out the strong semantics in a top-down manner. Our final model StairNet detector unifies the multi-scale representations and semantic distribution effectively. Experiments on PASCAL VOC 2007 and PASCAL VOC 2012 datasets demonstrate that StairNet significantly improves the weakness of SSD and outperforms the other state-of-the-art one-stage detectors.

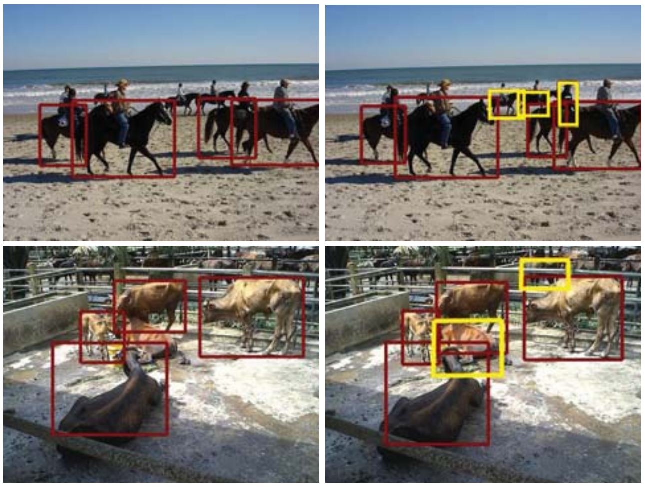

(a) (b)

Figure 1. **The Detection Output of original SSD and StairNet.** Given the input image, (a) shows the output of SSD and (b) shows the output of StairNet. As shown in the image, one-stage detectors [21, 27] have poor performance on objects that require significant context information(e.g. small, overlapped and truncated).

1. Introduction

Convolutional Neural Networks(CNNs) have significantly pushed the performance of visual recognition tasks such as image classification [12, 13, 17, 32, 34], semantic segmentation [3, 23, 26, 40] and object detection [9, 10, 21, 27, 30]. Thanks to the representation power of the CNNs, features learned by neural networks in an end-to-end, data-driven manner have provided dramatically better results than hand-crafted features. Hence, it is not surprising that most of the recent researches on visual recognition are based on network engineering rather than feature engineering [38]. Designing the better network architectures became a critical issues on a broad array of vision problems and among them object detection is one of the fastest-moving areas.

Recent object detectors that are based on the CNN can be divided into two streams. The first is two-stage detectors popularized by R-CNN [10], where sparse regions were proposed in the first stage then followed by a second stage for refinement. They guarantee high accuracy but is not suitable for real-time applications due to the high memory usage and slow speed, e.g 5FPS. On the other hand, the one-stage approaches such as the Single Shot Detector(SSD) [21] or You Only Look Once(YOLO) [27, 28] directly predict the output without region proposal module. They are fast and simple to train end-to-end. However, it is shown that they produce a low quality bounding boxes, hence, results in a failure localization of small objects or occluded objects. [29] We focus to alleviate this issues by carefully dissolving recent ideas into the network design.

Our first design principle is motivated by [7, 21]. Detecting objects of various scales has long been a demanding challenge. Prior to the advent of the CNNs, image pyramids were proposed as a solution. For example, deformable part models(DPM) [7] use a multi-scale images to produce a multi-scale features then the filters slide densely on top of the feature pyramid. Recent top-ranked detectors in the benchmarks [6, 20] also use multi-scale images for the training and testing. Despite the promising results of

978-1-5386-4886-5/18/$31.00 ©2018 IEEE
DOI 10.1109/WACV.2018.00125

image pyramids, computation time increases considerably and memory usage gets high. Instead, we draw on a recent approach [21], especially adopting the SSD-style pyramid. The multi-scale feature maps are already constructed by a subsampling layers in a deep CNN. SSD have shown the effectiveness of this cost-free inherent representations for object detection. However, the original SSD-style pyramid misses to exploit semantically strong information which is critical for small object detection. While following the SSD, we augmented the pyramid with our feature combining module to boost performance.

It has been widely recognized that the contextual information is decisive on detecting visually impoverished objects (e.g. small, truncated and occluded objects). Many of recent detectors are proposed to use contextual information. [1, 8, 19, 29]. However due to their complex [1, 29] and heavy architecture [8], the network shows slow inference time and is not able to be trained in end-to-end [8]. To address these issues, we propose a simple module that propagates semantically strong features, which contain contextual information, from top to bottom in the network. In this respect, FPN [19] is similar to ours in that the final goal is to construct a multi-scale feature maps with small semantic gaps. [19] combines high level features with low level features, enabled by a nearest neighbor upsampling and lateral connections. Instead, we adopt different components and carefully designed the top-down feature combining module that significantly improves the weakness of SSD. Our final model is called StairNet that unifies multi-scale representations and semantic distribution in an efficient way. StairNet is an end-to-end, one-stage detector which outperforms current state-of-the-art one-stage detectors.

Our main contributions are summarized as follows:

- We propose StairNet framework that effectively unifies multi-scale representations and semantic distribution.
- We conduct extensive ablation experiments and introduce a set of effective design choices for feature combining.
- We show that StairNet can acheive state-of-the-art performance on two standard benchmarks (PASCAL VOC 2007 and PASCAL VOC 2012) without losing real-time processing speed.

2. Related works

Object detection Detection frameworks have been dominated by sliding-window paradigm for many years. These methods heavily relied on hand-crafted features such as HOG [5]. However, after the dramatic performance boost brought on by R-CNN [10], which combines an object proposal mechanism [36] with a powerful CNN classifer, traditional methods were surpassed in a short period time. The R-CNN detector has been improved over the years both in terms of speed and accuracy. Recently, Faster-RCNN [30] integrated proposal generation module and the Fast R-CNN [9] classifier into a single CNN. Many researchers adopted [30] framework and proposed a numerous extensions. This two-stage approaches consistently have occupied the top entries of the challenging benchmarks so far. However, due to the *propose and classify* two-stage design, two-stage detector hurts the detection efficiency. They suffer from high memory usage and slow inference time. This motivates to build one-stage detectors that predicts outputs in a proposal-free manner.

OverFeat [31] is the first CNN based one-stage object detector using sliding-window paradigm. YOLO [27] and SSD [21] have recently been proposed for real-time detection. They are a fast single stage methods which divide an image in to a multiple grids and simultaneously predict bounding boxes and class confidences. Unlike YOLO, SSD uses in-network multiple feature maps to detect objects with sizes of a specified ranges. This makes SSD more robust to detect varying shapes and sizes of objects than YOLO. We adopt the SSD framework for our starting point.

Using multiple layers A number of studies have shown that exploiting multiple layers within a CNN can improve detection and segmentation. HyperNet [16] and ION [1] concatenate the features from different layers and pool object proposals from the coupled layer. FCN [23] and Hypercolumns [11] upsample multiple layers and combine partial scores of each layers for final decision. SSD [21] enforces each layer to predict certain scale of objects by distributing various scales of default boxes to multiple layers. Similar to SSD, MS-CNN [2] also uses multiple feature maps for prediction and they newly introduced deconvolution layer to increase the resolution of feature maps. FPN [19] attempted to leverage the pyramidal shape of CNN. They augmented the CNN to build strong semantics at all scales of feature maps, enabled by nearest neighbor upsampling and lateral connections.

Context information Global contexts are well known to play critical role in visual recognition problems. Recent architectures attempt to use this strong semantics for their specific tasks. DPM [7] integrated a global root model and finer local part models to represent deformable objects efficiently. Viewpoints and Keypoints [35] leverages the global viewpoint estimation to improve local keypoint predictions. RRC [29] transfers each feature semantic information to other layers by stacking pooling and deconvolution layers upon SSD. [8, 25, 26] have shown that encoder-decoder, hourglass shape, is effective to propagate the context information. FPN [19] builds rich semantics at all levels by combining each layers. CoupleNet [42] introduces global FCN branch to extract global semantics. All of which show that effective combination of the strong semantics(e.g. global context information) and fine local details improve the discrimination performance. Inspired by recent works, we propose to use top-down feature combining module to diffuse out the semantics effectively. Our proposed StairNet follows the SSD-style pyramid and thus it inherits the advantages of SSD, while produces more accurate models. We show that our model is simple and effective which outperforms current state-of-the art one-stage detectors.

3. StairNet

In this section, we begin by explaining the defect of the current detectors in details, then we elaborate the new improved detection framework, StairNet. We discuss why we choose the particular architecture and how we come up with our combining strategy.

3.1. A weak spot of current detectors

While recent CNN models designed for object detection have shown excellent capability to address the multiclass problem, less improvement has been made towards the detection of objects at various scales. For example, the Faster-RCNN [30] incorporated the proposal mechanism into single CNN. This saved computation time and enabled end-to-end training. However, for the object proposals, this approach only relies on the large receptive field of feature map(e.g. conv5). Since filter receptive fields are fixed but objects scales vary in natural scenes, this creates a discordance and compromises the performance. We can summarize this in a simple mathematical expression,

$$f_n = C_n(f_{n-1}) = C_n(C_{n-1}(...C_1(I))), \tag{1}$$

$$Object\ Proposals = P(f_n), \tag{2}$$

where I is an input image, C_n is a n_{th} convolution block that is composed of convolutional layers, pooling layers, ReLU layers, etc. f_n is the n_{th} layer feature map, P is the prediction layer that transforms certain feature map to detection ouputs: class confidence score and bounding box location.

Recently in order to resolve the problem of Eq. (2), SSD [21] and MS-CNN [2] focused on the fact that the internal feature maps of a deep CNN are already of multi-scale, pyramidal shape. They utilizes the low-resolution maps to detect large objects and high-resolution map to detect small objects. These two approaches can be expressed as follows,

$$\begin{gathered} Detection\ Outputs = \{P_{n-k}(f_{n-k}), ..., P_n(f_n)\}, \\ where\ n > k > 0 \end{gathered} \tag{3}$$

Since they directly enforce each layer to be responsible for certain scale, every layers that are used for prediction have to be semantically strong.

It is well known that SSD-style detectors have inferior performance on small objects while they are competitive with state-of-the-art two-stage detectors on large objects. [14] We conjecture that this is because the lower level feature maps do not contain strong semantics(e.g contextual information). Due to different receptive field sizes of each feature maps, they differ in the level of semantics they are containing. In other words, as the feature map level goes down, semantic level gradually decreases.($n \rightarrow n-k$) The lowest layer then contains weak semantics, local features. [41] found that the actual receptive field**(arf)** size is much smaller than the theoretical receptive field(**trf**) size. [24] shows that the pixels near the center of receptive field have much larger effect than the outer pixels, leading to a 2D-gaussian shape that also occupies smaller fraction than the trf. These findings indicate that the trf size sets an upper bound on the arf size. Since the **arf** size of f_{n-k} that is responsible for small objects in SSD is 58.6 [37], we can infer that f_{n-k} seriously lacks global context information and only sees the local part of image that has size of 300 or 512. In Eq. (3), it does not considers this problem well and directly uses f_{n-k} to detect small objects, leading to poor performance. Therefore, we propose a new effective expression to handle this problem as follow,

$$\begin{aligned} &Detection\ Outputs = \{P_{n-k}(f'_{n-k}), ..., P_n(f'_n)\} \\ &\qquad f'_n = f_n \\ &\qquad f'_{n-1} = f_n + f_{n-1} \\ &\qquad\quad \vdots \\ &\qquad f'_{n-k} = f_n + f_{n-1} + ... + f_{n-k}, \\ &\qquad\quad where\ n > k > 0 \end{aligned} \tag{4}$$

The Eq. (4) differs from Eq. (2) and Eq. (3), in that it uses multi-scale representations and distributes the stronger semantics gradually staring from the top layer. This gradual semantic aggregation is enabled by our iterative top-down feature combining operation. One thing to note that is still a one-stage process.

3.2. Network Architecture

The proposed StairNet is a single, unified network composed of 1) a meta-architecture(SSD), 2) the feature combining module, and 3) an unified prediction layer. Fig. 2 illustrates the architecture of proposed **StairNet** framework along with the original SSD. Our framework first processes an arbitrary single-scale image in a fully convolutional way. Then the feature combining module distributes the semantics through down the layer, starting from the top-most feature map that generally contains strong semantics. The enhanced multi-scale feature maps are then referenced by the unified classifier to output the final predictions. We show

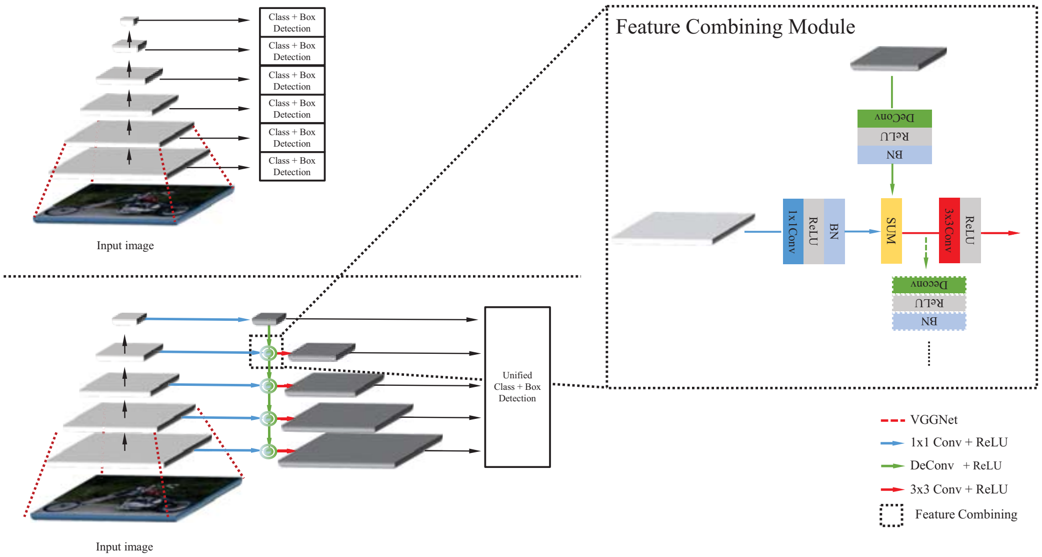


Figure 2. **StairNet** architecture augments the SSD with **feature combinining module**. The black dotted box shows the detail architecture of the feature combining module. This module spreads out the contextual information through down the layer effectively.

that our StairNet improves the abstraction level of lower layer effectively. We elaborate each component in the following.

3.2.1 Meta-architecture of StairNet

Our first design principle is to leverage an in-network feature pyramid, hence, we adopt the SSD framework as the (meta)-architecture of StairNet. In particular, within the SSD, the feature extractor can be substituted for recent off-the-shelf CNNs [12,13,33]. However in order to fairly measure the effectiveness of our proposed feature combining module, we maintained the feature extractor [32] identical to the original SSD.(Fig. 2)

Since the SSD contains multi-scale feature maps with a scaling step of 2, we take total five scales of feature maps that have strides of $\{8, 16, 32, 64, 100\}$ pixels with respect to the input images. We take first two of feature maps from the base network(conv4_3 and conv5_2 from VGGNet). Then the remaining three feature maps are selected from the output of the two-stride subsampling layers that are added after the base network. We adopt different default box scale distribution from the original SSD. Given five level of feature maps, we set the scale of default boxes to be $\{0.1, 0.2, 0.37, 0.54, 0.71\}$ respectively. We used the aspect ratio of default boxes to be $\{2, 3\}$ in all scales.

3.2.2 Feature Combining Module

Our second design principle is to make all feature maps semantically strong. We augment the conventional SSD with our feature combining module in order to propagate the high-level abstraction features of top layer to lower layer. (Fig. 2 black dotted box) illustrates the unit of feature combining module. It consists of three parts: 1x1 convolution layer, deconvolution layer and 3x3 convolution layer.

In order to combine the propagated information of upper layer and the original features of corresponding bottom layer, we introduce **1x1 convolution layers**. We adopt whole channels of feature maps to be 256 which is the minimum of original values$\{512, 1024, 512, 256, 256\}$ (Fig. 2 blue line). This choice is natural since it allows similar levels of influence when two feature maps are combined. To combine two different size of feature maps, we add **deconvolution layers** that upsample by a factor of 2 (Fig. 2 green line). The features of upper layer, that have more strong semantics relative to lower layer, are delivered by this deconvolution layer. Before combining them together, it is essential to normalize features from different layers since it shows different scale distribution [22]. Note that we use batch normalization to handle this problem. We then combine them using element-wise add operation. The combined features are passed down directly to the next deconvolution layer. The spectrum of information sources of each feature maps incrementally increases by this iterative process. The lower the feature map, the more the information sources are supplied to complement weak semantics of lower feature maps. Moreover, we also compare deconvolution layer with bilinear upsampling method and empirically verify that learnable upsampling-weights perform better (see Sec. 4.1.3). To effectively mix the information from different streams(Fig. 2 blue and green line), we apply a

3x3 convolution layer(Fig. 2 red line) to construct the final enhanced feature maps before the classifier. Note that the enhanced feature maps have same spatial sizes respect to the original feature maps. Instead of using 3x3 convolution layer, we additionally examine Resblock [12] which uses less parameter due to its bottleneck architecture ,however, we observe degraded performance and speed(1-2ms). We argue that its because skip connection passes noisy information(aliasing effect of upsampling) and small inner-product operations are less optimized in existing GPU libraries (see Sec. 4.1.3). Following pseudo code shows the overall operation of feature combining module.

Algorithm 1 Feature Combining Module

```
**INPUT**: **F** $= [F_1, F_2, F_3, F_4, F_5]$
**OUTPUT**: **O** $= [O_1, O_2, O_3, O_4, O_5]$

/*Feature Projection*/;
**for** t:=0 to N-1 do **do**
    $F_t \leftarrow P_t(F_t)$
**end for**

/*Feature Combination*/;
$O_{N-1} \leftarrow F_{N-1}$
**for** t:=N-2 to 0 do **do**
    $F_t \leftarrow F_t + \uparrow F_{t+1}$ /*up-arrow is up-sampling*/
    $O_t \leftarrow M_t(F_t)$
**end for**
```

3.2.3 Unified Prediction Layer

As shown in Table. 2 col7 and col8, despite of reducing the number of parameters by employing unified (shared) classifier, performance remains the same. Many previous works argued that object scale difference causes different distribution in feature space [18, 39]. Thus, like SSD [21], multiple classifiers for different scales have been popular choice for better performance. In this point of view, the result in Table. 2 implies that proposed feature combining module effectively mitigate the semantic gap between different hierarchical layers, i.e. the feature maps share similar degree of semantics in spite of scale difference. In this reason, by adopting the unified classifier we can reduce the parameters of classifier. Potential advantage of similar semantic representation over hierarchical layers and unified classifier is alleviation of training data imbalance problem over object scales. For the skewed distribution of a specific category, e.g. there are many large cows but few small cows in PASCAL VOC 2007, our method could be helpful because our module shares a single classifier over various scales: as the final feature representations of large cow and small cow are similar, the classifier trained with large cows would work well for small cows as well.

Extra ConvBlock	**SUM**	MAX	PRODUCT
w 3x3Conv	**78.8 (76.4)**	78.9 (76.2)	78.3
w/o 3x3Conv	78.2	77.8	-

Table 1. **Effects of Extra ConvBlock and Combining Methods** Metric: detection mAP(%) on VOC07 *test*. ( ): detection results of PASCAL VOC12 *test*.

	SSD							**StairNet**
{2, 3}		✓	✓		✓	✓	✓	✓
deconv			✓	✓		✓	✓	✓
3x3 conv				✓	✓		✓	✓
unified		✓	✓	✓	✓	✓		✓
{1.6, 2, 3}				✓				
bilinear					✓			
ResBlock						✓		
multi							✓	
VOC 2007 mAP	77.2	77.4	78.2	78.9	78.5	78.6	78.9	**78.8**

Table 2. **Ablation Experiments on StairNet.** Each component of the first row corresponds to each component of the second row in an one-to-one manner. $\{$**2,3**$\}$ **and** $\{$**1.6, 2, 3**$\}$ indicates the aspect ratio of the default boxes. **deconv and bilinear** indicates the upsampling method in the feature combining module. **3x3 conv and ResBlock** indicates the extra layers in the feature combining module. **unified and multi** indicates whether the prediction layer shares the weight or not.

Like the other one-stage classifiers, our unified prediction layer predicts the probability of object presence and bounding box offsets, at each spatial location for each of the k default boxes and c object classes. Taking the outputs of feature combing module as input, this prediction layer applies a 3x3 conv layer with $(c + 4)k$ filters.

4. Experiments

We evaluate the StairNet on the widely used datasets: PASCAL VOC 2007 and VOC 2012 benchmarks [6]. All of our experiments are based on the Pytorch framework. In order to better perform apple-to-apple comparisons, we first attempted to reproduce the original accuracy of SSD in the Pytorch framework and set as our baseline. Then we performed extensive ablation studies to thoroughly investigate the effectiveness of each component. Moreover, we evaluate our model on different object scales and verify that StairNet improvoes the weakness of SSD. Finally, we show that StairNet outperforms current state-of-the-art one-stage detectors.

4.1. Ablation studies on VOC2007

We perform ablation experiments on PASCAL VOC 2007 test sets for detailed analysis of our proposed StairNet framework. We train the models on the union set of VOC 2007 trainval and VOC2012 trainval (07+12), and evaluate on VOC 2007 test set. We first removed each com-

Method	data	network	map	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
HyperNet [16]	07+12	VGGNet	76.3	77.4	83.3	75.0	69.1	62.4	83.1	87.4	87.4	57.1	79.8	71.4	85.1	85.1	80.0	79.1	51.2	79.1	75.7	80.9	76.5
Fr R-CNN [30]	07+12	ResNet-101	76.4	79.8	80.7	76.2	68.3	55.9	85.1	85.3	89.8	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	85.3	72.0
ION [1]	07+12+S	VGGNet	76.5	79.2	79.2	77.4	69.8	55.7	85.2	84.2	89.8	57.5	78.5	73.8	87.8	85.9	81.3	75.3	49.7	76.9	74.6	85.2	82.1
R-FCN [4]	07+12	ResNet-101	80.5	79.9	87.2	81.5	72.0	69.8	86.8	88.5	89.8	67.0	88.1	74.5	89.8	90.6	79.9	81.2	53.7	81.8	81.5	85.9	79.9
SSD300* [21]	07+12	VGGNet	**77.2**	82.3	84.5	75.0	69.9	51.2	85.2	85.6	87.5	63.0	82.6	76.2	84.2	86.5	83.8	78.6	51.0	75.1	79.6	86.7	75.5
SSD300 [21]	07+12	VGGNet	77.5	79.5	83.9	76.0	69.6	50.5	87.0	85.7	88.1	60.3	81.5	77.0	86.1	87.5	83.9	79.4	52.3	77.9	79.5	87.6	76.8
DSSD321 [8]	07+12	ResNet-101	78.6	81.9	84.9	80.5	68.4	53.9	85.6	86.2	88.9	61.1	83.5	78.7	86.7	88.7	86.7	79.7	51.7	78.0	80.9	87.2	79.4
StairNet	07+12	VGGNet	**78.8**	81.3	85.4	77.8	72.1	59.2	86.4	86.8	87.5	62.7	85.7	76.0	84.1	88.4	86.1	78.8	54.8	77.4	79.0	88.3	79.2

*: reproduced in PyTorch framework.

Table 3. **PASCAL VOC 2007 *test* detection results. 07+12**: 07 trainval + 12 trainval. **07+12+S**: 07+12 plus segmentation labels.

Method	data	network	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
HyperNet [16]	07++12	VGGNet	71.4	84.2	78.5	73.6	55.6	53.7	78.7	79.8	87.7	49.6	74.9	52.1	86.0	81.7	83.3	81.8	48.6	73.5	59.4	79.9	65.7
Fr R-CNN [30]	07++12	ResNet-101	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
ION [1]	07++12+S	VGGNet	76.4	87.5	84.7	76.8	63.8	58.3	82.6	79.0	90.9	57.8	82.0	64.7	88.9	86.5	84.7	82.3	51.4	78.2	69.2	85.2	73.5
RFCN m-sc [4]	07++12	ResNet-101	77.6	86.9	83.4	81.5	63.8	62.4	81.6	81.1	93.1	58.0	83.8	60.8	92.7	86.0	84.6	84.4	59.0	80.8	68.6	86.1	72.9
YOLOv2 [28]	07++12	Darknet-19	73.4	86.3	82.0	74.8	59.2	51.8	79.8	76.5	90.6	52.1	78.2	58.5	89.3	82.5	83.4	81.3	49.1	77.2	62.4	83.8	68.7
SSD300* [21]	07++12	VGGNet	**74.8**	87.7	83.4	73.0	60.3	47.7	80.6	76.7	91.5	57.6	77.6	63.6	89.0	84.9	84.8	81.8	48.9	77.9	72.3	86.1	71.2
SSD300 [21]	07++12	VGGNet	75.8	88.1	82.9	74.4	61.9	47.6	82.7	78.8	91.5	58.1	80.0	64.1	89.4	85.7	85.5	82.6	50.2	79.8	73.6	86.6	72.1
DSSD321 [8]	07++12	ResNet-101	76.3	87.3	83.3	75.4	64.6	46.8	82.7	76.5	92.9	59.4	78.3	64.3	91.5	86.6	86.6	82.1	53.3	79.6	75.7	85.2	73.9
StairNet	07++12	VGGNet	**76.4**	87.7	83.1	74.6	64.2	51.3	83.6	78.0	92.0	58.9	81.8	66.2	89.6	86.0	84.9	82.6	50.9	80.5	71.8	86.2	73.5

*: reproduced in PyTorch framework.

Table 4. **PASCAL VOC 2012 *test* detection results. 07++12**: 07 trainval + 07 test + 12 trainval. **07+12+S**: 07+12 plus segmentation labels. Result link of **StairNet** : http://host.robots.ox.ac.uk:8080/anonymous/SPPVPF.html

ponent step-by-step to observe real effects of each component.(Table. 2 cols 9,4,3 and 2) We then conduct controlled experiment to investigate several design choices in our model.(Table. 2 cols 9,8,7,6 and 5)) In all the experiments, the size of input image is fixed to 300 for simplicity. The results are mainly summarized in Table. 1 and Table. 2.

4.1.1 Combining methods: Element-wise operation

Element-wise sum We fist investigate three different combining methods: element-wise sum, element-wise product and element-wise max. (Table. 1 row 2) shows that using element-wise sum generates the best performance. This phenomenon can be interpreted in terms of information flow. The ResidualNet [12], which achieves state of the art results in many challenging vision tasks, shows that the element-wise sum is effective way to integrate and preserve the information. In forward phase, it enables network to use the information from two branches complementary without losing any of information. In the backward phase, the gradient is distributed equally to all of inputs, leading to an efficient training. The element-wise maximum, which routes the gradient only to the higher inputs provides regularization effect in some extent, it generates unstable performance. The element-wise product, which assigns a small gradient to the large input and a large gradient to the small input leads network hard to converge, yielding the worst performance. Therefore, we use element-wise sum for the following experiments.

4.1.2 Analysis of each component in StairNet

Extra ConvBlock As shown in Table. 2 of col 4, we observe performance drop without the 3x3 convolution layer. In order to investigate the role of 3x3 convolution layer thoroughly, we conducted additional experiment shown in Table. 1. We observe that without the 3x3 convolution layer, performances are consistently degraded regardless of the type of combining methods. In the case of element-wise product, the training was even not stable without the 3x3 convolution layer. This experiment shows that the extra 3x3 convolution layer not only improves the performance but also helps training more stable. Since introducing 3x3 convolution layer makes the total depth of the model more deep, it increases the capacity of the model. Moreover, we conjecture that 3x3 convolution layer acts like a buffer that constructs similar semantic levels of final feature maps before the unified classifier.

Top-bottom connection As shown in Table. 2 of col 3, we observe significant drop of performance without the deconvolution layer, i.e. removing top-bottom connection. This shows that top-down semantic aggregation plays a critical role in feature combining module. The deconvolution layer helps the distribution of semantics from top to bottom and reduces the semantic gap of feature maps. We can also adopt naive upsampling methods such as nearest neighbor and bilinear interpolation. [19] However we found deconvolution layer performs better than simple upsampling method. We will discuss this point in the following section.

4.1.3 Model architecture design

Recently [8] conducted k-means clustering on the bounding boxes in the training data of PASCAL VOC 2007 and VOC 2012 and they included one more aspect ratio of 1.6 to improve the performance. However, as shown in (Table. 2 cols 5 and 9) we observe no significant improvement with it. Since using less aspect ratio not only saves weight parameters but also computation times, we stick to {2,3} aspect ratio.

Method	scale	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
SSD300* [21]	small	41.7	47.2	64.7	35.2	23.2	8.5	45.7	49.6	67.2	20.0	47.6	30.1	53.1	61.6	46.9	27.8	7.3	44.5	62.0	63.3	28.3
StairNet	small	**50.6**	57.5	71.6	51.4	34.5	21.8	54.2	56.2	65.4	24.0	73.2	31.6	60.4	73.3	54.1	36.7	21.5	51.7	59.5	70.0	43.8
SSD300* [21]	medium	76.8	79.9	81.6	75.6	65.0	44.4	88.0	86.5	87.3	66.2	83.5	74.9	83.6	87.7	86.5	80.3	46.5	75.2	78.9	86.1	77.0
StairNet	medium	**78.0**	79.1	83.1	77.9	69.2	55.0	88.1	86.7	88.0	62.5	83.1	78.0	83.1	89.3	88.5	79.1	49.7	74.5	79.4	87.7	78.5
SSD300* [21]	large	**80.4**	90.5	88.1	80.1	80.5	63.7	88.3	90.2	90.7	50.9	82.6	84.9	84.8	87.8	90.1	85.5	57.7	82.1	62.4	88.3	78.8
StairNet	large	78.5	90.4	89.3	85.8	76.2	63.5	92.0	89.4	88.4	47.2	81.8	80.3	80.5	88.2	89.8	83.5	52.8	76.6	46.5	89.9	79.0

*: reproduced in PyTorch framework.

Table 5. **Scale-aware evaluation on PASCAL VOC 2007 *test*.** Two methods are trained on VOC 07+12. VGGNet is used as backbone in both methods.

Method	data	network	mAP	fps	lib
YOLO [28]	07+12	GoogLeNet	63.4	45	DarkNet
YOLOV2_352 [28]	07+12	DarkNet-19	73.7	81	DarkNet
YOLOV2_544 [28]	07+12	DarkNet-19	78.6	40	DarkNet
SSD300* [21]	07+12	VGGNet	77.2	42	PyTorch
SSD300 [21]	07+12	VGGNet	77.5	62	Caffe
DSSD321 [8]	07+12	ResNet-101	78.6	9.5	Caffe
RSSD300 [15]	07+12	VGGNet	78.5	35.0	Caffe
DiCSSD300 [37]	07+12	VGGNet	78.1	40.8	Caffe
StairNet	07+12	VGGNet	**78.8**	30	PyTorch

*: reproduced in PyTorch framework.

Table 6. **PASCAL VOC 2007 *test* detection results.** Is is worth to note that PyTorch implementation runs slower (62fps → 42fps) and shows lower performance (77.5→77.2) than Caffe implementation for exactly same algorithm (Row 4 & 5). In spite of this disadvantage, **StairNet** outperforms other state-of-the-art methods.

		Recall						
Method	data	0.5	0.6	0.7	0.8	0.9	1	mAP@0.7+
SSD300* [21]	07+12	91.9	87.9	79.7	65.6	34.4	0	44.9
R-SSD300 [15]	07+12	92.7	88.4	82.4	68.9	37.6	0	47.2
StairNet	07+12	**94.3**	**90.1**	**83.5**	**70.1**	**38.8**	0	**48.1**

*: reproduced in PyTorch framework.

Table 7. **mAP at Recall≥0.7** mean average precision over recall≥0.7 suggested by [15]. In most practical cases, it is more important to achieve high-precision at high-recall range rather than at low-recall range. Our **StairNet** outperforms SSD and R-SSD.

Generally, there are two ways to enlarge the resolution(x2) of feature map. First, the deconvolution layer that their upsampling weights are learned through the training process. Second, naive upsampling methods such as nearest neighbor or bilinear interpolation. As can be seen in (Table. 2 cols 6 and 9) deconvolution layer improves the performance. This implies that the learned upsampling-weights perform better than the naive upsampling kernels. Moreover, the recent studies [8,26] have shown that the sequence of deconvolution layer is suitable for propagating the information efficiently.

We already have shown the effectiveness of 3x3 convolution layer in Sec. 4.1.2. Rather than 3x3 convolution layer, we experiment with ResBlock to make each detection path deeper. The detail architecture of ResBlock[1] is explained in supplemental section. As shown in the (Table. 2 cols 7 and 9), we observe that 3x3 convolution layer performs better than ResBlock. We conjecture that since the combined features which are fed into ResBlock would have redundant information, skip path in ResBlock would deliver unnecessary information which causes performance degradation.

Unlike original SSD, our final StairNet uses unified classifier. As shown in (Table. 2 cols 8 and 9) we observe no big difference on performance which indicates that all feature maps share similar degree of semantics. This justifies the effectiveness of our feature combining module that spreads out the information effectively. We adopt unified classifier for our final model to save weight parameters.

4.2. Impact on different sized objects

To analyze the impact of our final model on the detection performance of different sized objects, we evaluated SSD and StairNet, considering objects of three different sizes. We find that the MS COCO criteria for object scales causes serious imbalance on VOC2007 test set, leading to undesirable comparison. We show this statistics in supplemental section. Instead, for each class we sorted ground truth bounding boxes on test set by area and divided them into three part: $small : [\sim 25\%)$, $medium : [25\% \sim 75\%)$ and $large : [75\% \sim]$ which consistently results in 1:2:1 ratio of number of sizes of test set for each class. When benchmarking on objects of each size, the ground truth labels for other sizes were ignored. As shown in Table. 5, proposed method shows significantly better performance than SSD on small scales.(8.9 mAP increase) StairNet wins on 18 classes among 20 classes. Even though the non-rigid objects such as cow, horse, person, and bird look very different due to its deformability, StairNet works better on these categories because it captures contextual information.

4.3. PASCAL VOC 2007 Results

We trained our model on the union of 2007 trainval and 2012 trainval. We used the same training scheme for both SSD and StairNet. We used a weight decay of 0.0001 and a momentum of 0.9. A batch size was set to 16 and adopted SGD optimizer with initial learning rate 0.001. We then decreased it by a factor of 10 at 80K and 100K iterations respectively. The training was terminated at 120K iterations.

Table. 3 shows our results on the PASCAL VOC 2007 test set. SSD* is the reproduced version in Pytorch framework by ourselves and we achieved 77.2 %. StairNet achieves a mAP of 78.8 %, which outperforms the SSD by 1.6 points. Our model even outperforms the DSSD which

[1] Standard basic block: [(C1-R)+(C1-R-C3-R-C1)]-Sum-R

uses ResNet-101 as their base network. Note that our StairNet model shows a large improvement over the classes with specific backgrounds like boat, car, cow, train, i.e. water for boat and railroad for train and so on. We also observed significant gain over the objects that mainly contain a small sizes of ground truth boxes such as bottle and plant.

We evaluate the inference time of our network using a NVIDIA-TITAN X GPU (pascal) along with CUDA 8.0 and cuDNN-v5.1. As shown in Table. 6, StairNet outperforms all the current one-stage methods in 30fps. One thing to note is that Pytorch implementation runs slower and shows lower perfomance than original Caffe implementation for exactly same algorithm. Inspite of this disadvantage, StairNet outperforms other state-of-the-art methods including most recent SSD-based detectors. [8, 15, 37]

Moreover we evaluate our model on mean average precision over Recall$\geq$0.7 suggested by [15]. Table. 7 shows the results that even in the high-recall range our model achieves high-precision and outperforms SSD and R-SSD [15].

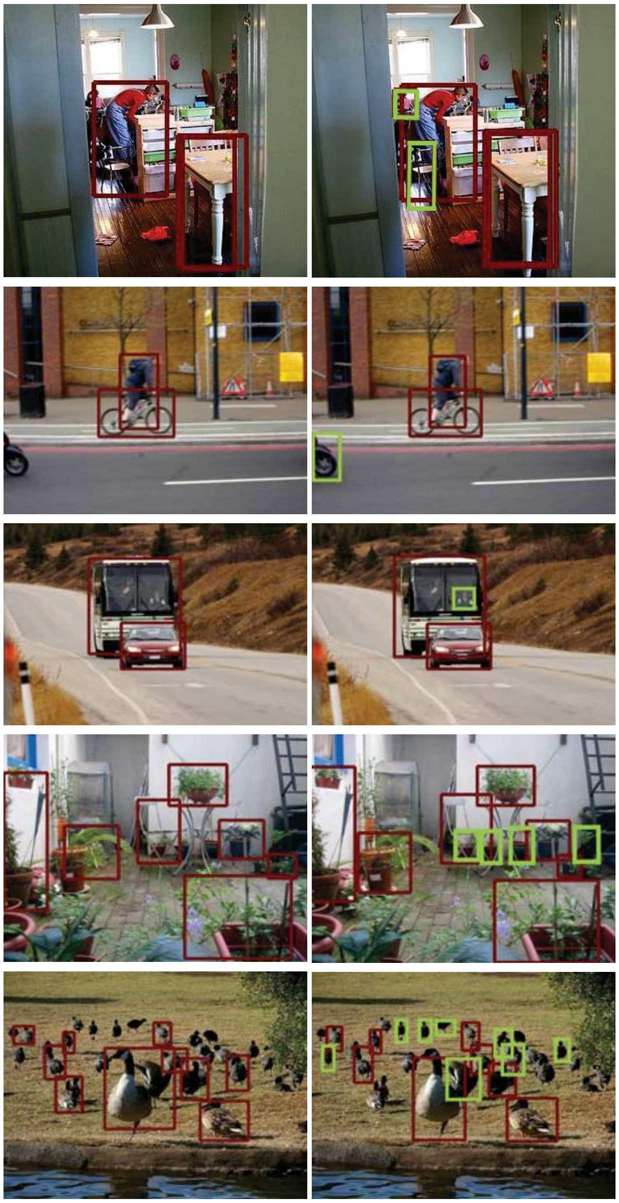
4.4. PASCAL VOC 2012 Results

We also evaluate our method on the more challenging VOC2012 dataset by submitting results to the public evaluation server. We use VOC 2007 test, VOC 2007 trainval and VOC2012 trainval as the training set. We follow the same setting of VOC2007 except the number of total iterations. Since there are more training images we increased the number of training iterations to 140K. Starting from the same learning rate of 0.001, we then decreased it by a factor of 10 at 80k, 100k and 120k iterations respectively. The training was terminated at 140K iterations.

Table. 4 shows the results on the VOC2012 test set. Our method achieves 76.4% mAP, which outperforms the SSD by 1.6%. As shown in the table we observe similar improvement over the specific class. The StairNet outperforms all other one-stage methods once again.

5. Conclusion

In this paper, we present the StairNet an effective one-stage detector that spreads out the strong semantics in a top-down manner and constructs an enhanced multi-scale feature maps for accurate detection. We point out that two-stage methods do not utilize the advantages of inherent multi-scale feature maps while one-stage methods ignore to incorporate global context information. To address this, we augment the SSD framework with our feature combing module, leading to significant improvement on detecting small objects. The StairNet is simple and fast. We verify its efficacy by showing that it achieves state-of-the-art accuracy on two standartd benchmarks.



(a) SSD300*

(b) StairNet300

Figure 3. **Qualitative results on PASCAL VOC 2007**. Green boxes indicate that **StairNet** performs better than SSD in challenge scenarios.

Acknowledgment

This work is supported by two projects (1) Development of core technology for advanced locomotion/manipulation based on high-speed/power robot platform and robot intelligence(No.10070171) and (2) Development of Autonomous Emergency Braking System for Pedestrian Protection project(No.10044775). These projects are funded by the Ministry of Trade, Industry and Energy of Korea (MOTIE).

References

[1] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 6

[2] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2016. 2, 3

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *Proc. of Int'l Conf. on Learning Representations (ICLR)*, 2015. 1

[4] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Proc. of Neural Information Processing Systems (NIPS)*, 2016. 6

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2005. 2

[6] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. In *Int'l Journal of Computer Vision (IJCV)*, volume 88, pages 303–338, 2010. 1, 5

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. In *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, volume 32, pages 1627–1645, 2010. 1, 2

[8] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017. 2, 6, 7, 8

[9] R. Girshick. Fast r-cnn. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 2

[11] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 4, 5, 6

[13] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 4

[14] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[15] J. Jeong, H. Park, and N. Kwak. Enhancement of ssd by concatenating feature maps for object detection. In *BMVC*, 2017. 7, 8

[16] T. Kong, A. Yao, Y. Chen, and F. Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 6

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. of Neural Information Processing Systems (NIPS)*, 2012. 1

[18] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan. Scale-aware fast r-cnn for pedestrian detection. In *arXiv preprint arXiv:1510.08160*, 2015. 5

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6

[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. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2014. 1

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2016. 1, 2, 3, 5, 6, 7

[22] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015. 4

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2

[24] W. Luo, Y. Li, R. Urtasun, and R. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Proc. of Neural Information Processing Systems (NIPS)*, 2016. 3

[25] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2016. 2

[26] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 7

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2

[28] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 6, 7

[29] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu. Accurate single stage detector using recurrent rolling convolution. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. of Neural Information Processing Systems (NIPS)*, 2015. 1, 2, 3, 6

[31] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *Proc. of Int'l Conf. on Learning Representations (ICLR)*, 2013. 2

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of Int'l Conf. on Learning Representations (ICLR)*, 2014. 1, 4

[33] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. 4

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[35] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[36] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. In *Int'l Journal of Computer Vision (IJCV)*, volume 104, pages 154–171, 2013. 2

[37] W. Xiang, D.-Q. Zhang, V. Athitsos, and H. Yu. Context-aware single-shot detector. *arXiv preprint arXiv:1707.08682*, 2017. 3, 7, 8

[38] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[39] J. Yan, X. Zhang, Z. Lei, S. Liao, and S. Z. Li. Robust multi-resolution pedestrian detection in traffic scenes. In *CVPR*, 2013. 5

[40] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *Proc. of Int'l Conf. on Learning Representations (ICLR)*, 2016. 1

[41] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *Proc. of Int'l Conf. on Learning Representations (ICLR)*, 2015. 3

[42] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu. Couplenet: Coupling global structure with local parts for object detection. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2017. 2