Triply Supervised Decoder Networks for Joint Detection and Segmentation

Jiale Cao¹, Yanwei Pang¹*, Xuelong Li²
¹School of Electrical and Information Engineering, Tianjin University
²Center for OPTical IMagery Analysis and Learning, Northwestern Polytechnical University
connor@tju.edu.cn, pyw@tju.edu.cn, li@nwpu.edu.cn

Abstract

Joint object detection and semantic segmentation is essential in many fields such as self-driving cars. An initial attempt towards this goal is to simply share a single network for multi-task learning. We argue that it does not make full use of the fact that detection and segmentation are mutually beneficial. In this paper, we propose a framework called TripleNet to deeply boost these two tasks. On the one hand, to deeply join the two tasks at different scales, triple supervisions including detection-oriented supervision and class-aware/agnostic segmentation supervisions are imposed on each layer of the decoder. Class-agnostic segmentation provides an objectness prior to detection and segmentation. On the other hand, to further intercross the two tasks and refine the features in each scale, two light-weight modules (i.e., the inner-connected module and the attention skip-layer fusion) are incorporated. Because segmentation supervision on each decoder layer is not performed at the test stage and two added modules are light-weight, the proposed TripleNet can run at a real-time speed (16 fps). Experiments on the VOC 2007/2012 and COCO datasets show that TripleNet outperforms all the other one-stage methods on both two tasks (e.g., 81.9% mAP and 83.3% mIoU on VOC 2012, and 37.1% mAP and 59.6% mIoU on COCO) by a single network.

1. Introduction

Object detection and semantic segmentation are two fundamental tasks in the field of computer vision. Most state-of-the-art methods merely focus on one single task (i.e., object detection [42, 31, 28, 4] or semantic segmentation [24, 45, 14, 18]). However, simultaneous object detection and semantic segmentation is very important in many applications, such as self-driving cars and unmanned surface vessels. Thus, joint detection and segmentation is necessary because simply using two state-of-the-art networks for detection and segmentation is inefficient and time-consuming.

Moreover, object detection and semantic segmentation are highly related. On the one hand, semantic segmentation used as a multi-task supervision can provide context information and semantic features for object detection [35, 25]. On the other hand, object detection can be used as an object prior knowledge to help improve the performance of semantic segmentation [14, 40, 39].

Due to the application requirement and task relevance, joint object detection and semantic segmentation is very necessary, which has attracted the attentions of researchers. Fig. 1 summarizes three typical architectures of joint detection and segmentation. Fig. 1(a) shows the most naive way where two branches of object detection and semantic segmentation are in parallel attached to the last layer of the encoder [2]. In Fig. 1(b), the branch for object detection is further refined by the features from the branch for semantic segmentation [35, 53]. Recently, the encoder-decoder network is further proposed for joint detection and segmentation. In Fig. 1(c), each layer of the decoder is used for multi-scale object detection, and the concatenated...
feature maps from the different layers of the decoder are used for semantic segmentation [8]. The above methods have achieved great success on detection and segmentation. However, the performance is still far from the strict demand of real applications. We argue that these methods simply sharing a single network for multi-task learning do not fully exploit the mutual benefit between the two tasks.

To deeply exploit the mutual benefits for joint object detection and semantic segmentation, we propose a new framework called TripleNet in this paper (see Fig. 1(e)). On the one hand, to deeply join the two tasks at different scales, three types of supervisions (i.e., detection-oriented supervision, class-aware/agnostic segmentation supervisions) are imposed on each layer of the decoder network. On the other hand, to further intercross the two related tasks and refine the decoder features, two light-weight modules (i.e., the inner-connected module and attention skip-layer fusion) are incorporated. As a simplified version of TripleNet, PairNet in Fig. 1(d) is also proposed, which only imposes the detection supervision and class-aware segmentation supervision on each layer of the decoder. Finally, the contributions of this paper can be summarized as follows:

1. Two novel frameworks (i.e., PairNet and TripleNet) are proposed to deeply join detection and segmentation. In TripleNet, the detection-oriented supervision and class-aware/agnostic segmentation supervisions are imposed on each layer of the decoder. Meanwhile, two light-weight modules (i.e., inner-connected module and attention skip-layer fusion) are also incorporated in each decoder layer.

2. The synergies in both accuracy and speed are gained. On the one hand, both the detection and segmentation accuracies are significantly improved. On the other hand, TripleNet can run at a real-time speed because the improvement is not at expense of the extra computational costs.

3. Experiments on the VOC 2007, VOC 2012, and COCO datasets [9, 29] demonstrate the effectiveness and efficiency of the proposed TripleNet. For example, using a single network, it achieves 81.9% mAP and 83.3% mIoU on VOC 2012 without COCO pretraining at 16 fps.

2. Related works

Object detection It aims to classify and locate objects in an image. Generally, the methods can be divided into two main classes: two-stage methods and one-stage methods.

Two-stage methods firstly extract some candidate object proposals from an image and then classify these proposals into the specific object categories. R-CNN [12] and its variants (e.g., Fast RCNN [11] and Faster RCNN [42]) are the most representative frameworks among the two-stage methods. Based on R-CNN series, researchers have done many improvements [7, 27, 3]. To accelerate detection speed, Dai et al. [7] proposed R-FCN which uses position-sensitive feature maps for proposal classification and bounding box regression. To output multi-scale feature maps with strong semantics, Lin et al. [27] proposed feature pyramid network (FPN) based on skip-layer connection and top-down pathway. Recently, Cai et al. [3] trained a sequence of detectors with increasing IoU thresholds to improve detection quality.

One-stage methods directly predict object class and bounding box in a single network. YOLO [41] and SSD [31] are two earliest one-stage methods. After that, many variants were proposed [10, 23, 44, 55, 37]. DSSD [10] and RON [23] use the encoder-decoder network to add context information for multi-scale object detection. To train the object detector from scratch, DSOD [44] uses the dense layer-wise connections on SSD for deep supervision. Instead of using the in-network feature maps of different resolutions for multi-scale object detection, STDN [55] uses scale-transferable module to generate the different high-resolution feature maps based on the last layer. To solve class imbalance, RetinaNet [28] introduces focal loss to down-weight the contribution of easy samples.

Semantic segmentation It aims to predict the semantic label of each pixel in an image, which has achieved the significant progress with Fully Convolutional Networks (i.e., FCN [34]). Generally, the methods of semantic segmentation can also be divided into two main classes: encoder-decoder methods and spatial pyramid methods.

The encoder-decoder methods contain two subnetworks: an encoder subnetwork and a decoder subnetwork. The encoder subnetwork extracts strong semantic features and reduces spatial resolution of feature maps, which is usually based on the deep CNN models (e.g., VGG [45], ResNet [16], DenseNet [18]) pre-trained on ImageNet [43]. The decoder subnetwork gradually upsamples the feature maps of encoder subnetwork. For example, DeconvNet [36] and SegNet [1] use max-pooling indexes to upsample the feature maps. To extract context information, some methods [38, 26, 50] adopt the skip-layer connection to combine the feature maps of the encoder and decoder subnetworks.

Spatial pyramid methods adopt the idea of spatial pyramid pooling [15] to extract multi-scale information from the feature maps of last layer. Chen et al. [5, 6, 47, 48] proposed to use multiple convolutional layers of different atrous rates in parallel (called ASPP) to extract multi-scale features. Instead of using the convolutional layers of different atrous rates in parallel, Zhao et al. [54] proposed pyramid pooling module (called PSPNet), which downsamples and upsamples the feature maps in parallel. Yang et al. [48] proposed to use dense connection to densely cover object scale range.

Joint object detection and semantic segmentation It aims to simultaneously detect objects and predict pixel semantic labels by a single network. Recently, researchers have made some attempts. Yao et al. [49] proposed to use the graphical model to understand holistic scene. Teichmann et al. [46] proposed to join object detection and
Figure 2. The proposed PairNet for joint object detection and semantic segmentation. (a) The detailed architecture of PairNet. Each layer of the decoder is simultaneously used for detection and segmentation. (b) The skip-layer fusion used in PairNet. Please note only the segmentation in gray rectangle is used for final segmentation during inference.

recent years, Fully Convolutional Networks (FCN [34]) with encoder-decoder structure have achieved the immense success on object detection [28, 10] and semantic segmentation [1]. For example, DSSD [10, 40] and RetinaNet [28] use the different layers of the decoder to detect the objects of different scales. By using the encoder-decoder structure, SegNet [1] and LargeKernel [38] generate the high-resolution logits for semantic segmentation. Based on the above observations, a very natural and simple idea is that FCN with the encoder-decoder is suitable for joint object detection and semantic segmentation.

Based on FCN with the encoder-decoder, the paired supervision decoder network (i.e., PairNet) and triply supervised decoder network (i.e., TripleNet) for joint object detection and semantic segmentation are proposed.

3.1. Paired supervision decoder network (PairNet)

Based on the encoder-decoder structure, a very simple PairNet is firstly proposed. The supervisions of detection and segmentation are added at each decoder layer to guide feature learning at different scales. On the one hand, PairNet uses different layers of the decoder to detect objects of different scales. On the other hand, instead of using the last high-resolution layer for semantic segmentation which is adopted by most state-of-the-art methods [1, 38], PairNet uses each layer of the decoder to respectively parse pixel semantic labels. Though the proposed PairNet is very simple and naive, it has not been explored for joint object detection and semantic segmentation to the best of our knowledge.

Fig. 2(a) gives the detailed architecture of PairNet. The input image firstly goes through a fully convolutional network with encoder-decoder structure. The encoder gradually down-samples the feature maps. In this paper, ResNet [16] (i.e., res1-res4) and some new added residual blocks (i.e., res5-res7) construct the encoder. The decoder gradually maps the low-resolution feature maps to the high-resolution feature maps. To enhance context information, skip-layer fusion is used to combine the feature maps from the decoder and the corresponding feature maps from the encoder. Fig. 2(b) gives the illustration of skip-layer fusion. The feature maps in the encoder is firstly upsampled by bilinear interpolation and then concatenated with the corresponding feature maps of the same resolution in the encoder. After that, the concatenated feature maps go through a residual unit to generate the output feature maps.

To join object detection and semantic segmentation, each layer of the decoder is split into two different branches. The branch of object detection consists of a $3 \times 3$ convolutional layer and two sibling $1 \times 1$ convolutional layers for object classification and bounding box regression. The branches of object detection at different layers are used to detect objects of different scales. Specifically, the branch at front
layer of the decoder with low-resolution maps is used to detect large-scale objects, while the branch at latter layer with high-resolution maps is to detect small-scale objects.

The branch of semantic segmentation consists of a $3 \times 3$ convolutional layer to generate the logits. There are two different ways to generate the logits. The first is that the segmentation logits are upsampled to the same resolutions of the ground-truths, and the second is that the ground-truths are downsampled to the same resolutions of the logits. We find that the first strategy has a slightly better performance, which is adopted as follows.

### 3.2. Triply supervised decoder network (TripleNet)

To deeply exploit the mutual benefits between the two tasks, triply supervision decoder network (called TripleNet) is further proposed. On the one hand, to deeply join two tasks at different scales, detection-oriented supervision, class-aware/agnostic segmentation supervisions are added on each layer of the decoder. On the other hand, to further intercross the two tasks and refine the features in each decoder layer, the inner-connected module and attention skip-layer fusion are proposed. Fig. 3(a) gives the detailed architecture of TripleNet. Compared to PairNet, TripleNet adds some new modules as follows.

**Multi-scale fused segmentation** In [8, 54, 5], it has been demonstrated that multi-scale features can extract context information for semantic segmentation. To use multi-scale features for better semantic segmentation, the feature maps of different layers in the decoder go through a $3 \times 3$ convolutional layer for channel reduction. The output feature maps are upsampled to the same spatial resolution and concatenated together. After that, a $3 \times 3$ convolutional layer is used to generate the segmentation logits. Compared to the features from one layer of the decoder, multi-scale fused features make better use of context information. Thus, multi-scale fused segmentation is used for final prediction during inference. Meanwhile, the semantic segmentation on each layer of the decoder can be seen as a deep supervision for feature learning.

**The inner-connected module** PairNet only shares each layer of the decoder for detection and segmentation, while the branches of detection and segmentation in each layer of PairNet have no cross. To further intercross the two tasks, an inner-connected module is proposed. Fig. 3(b) shows the inner-connected module in layer $i$. The feature map in layer $i$ first goes through a $3 \times 3$ convolutional layer to produce the segmentation logits for the branch of semantic segmentation. Meanwhile, the segmentation logits further go through two $3 \times 3$ convolutional layers to generate the new feature maps which are concatenated with the feature maps in layer $i$. Based on the concatenated feature maps, a $3 \times 3$ convolutional layer is used to generate the feature maps for the branch of object detection.

**Class-agnostic segmentation supervision** Semantic segmentation mentioned above is class-aware, which aims to simultaneously identify specific object categories and the background. We argue that class-aware semantic segmentation may ignore the discrimination between objects and the background. Therefore, class-agnostic segmentation supervision is further added to each layer of the decoder. Specifically, a $3 \times 3$ convolutional layer is used to generate the logits of class-agnostic semantic segmentation. To generate the ground-truths, all the objects are set as one category, and the background is set as another category.

**Attention skip-layer fusion** In Section 3.1, PairNet simply fuses the feature maps of the decoder and the encoder. Generally, the features from the layer of the encoder have relatively low-level semantic, and that from the layer of the decoder have relatively high-level semantic. To enhance informative features and suppress less useful features from the
encoder by the decoder, Squeeze-and-Excitation (SE) [17] block in Fig. 3(c) is incorporated. The input of a SE block is the feature maps from the decoder, and the output of SE block is used to scale the feature maps from the encoder. After that, the feature maps from the encoder and decoder layers are concatenated for skip-layer fusion as PairNet.

4. Experiments

4.1. Datasets and evaluation

To demonstrate the effectiveness and efficiency of proposed methods and compare with some state-of-the-art methods, the experiments are conducted on the famous VOC 2007, VOC 2012 [9], and COCO [29] datasets.

The PASCAL VOC challenge [9] has been held annually since 2006 and consists of three principal challenges (i.e., image classification, object detection, and semantic segmentation). Among these annual challenges, the VOC 2007 and VOC 2012 datasets are usually used to evaluate the performance of object detection and semantic segmentation, which have 20 object categories. The VOC 2007 dataset contains 5011 trainval images and 4952 test images. The VOC 2012 dataset is split into three subsets (i.e., train, val, and test). The train set contains 5717 images for detection and 1464 images for segmentation (called VOC12-train-seg). The val set contains 5823 images for detection and 1449 images for segmentation (called VOC12-val-seg). The test set contains 10991 images for detection and 1456 for segmentation. To enlarge the training data for semantic segmentation, the augmented segmentation data provided by [13] is used, which contains 10582 training images (called VOC12-trainvalaug-seg).

The COCO benchmark [29] is a large-scale dataset for object detection, instance segmentation, and image caption, which has 80 object categories. It is usually split into three subsets (i.e., trainval35k, minival, and test). The trainval35k set contains about 115k images, the minival set contains 5k images, and the test set contains about 20k images. Usually, the trainval35k set is used for training, the minival set is for validation experiments, and the test set is for comparison with state-of-the-art methods. For semantic segmentation, the ground-truths are generated by assigning the objects which belong to the same category with the same semantic label.

**Evaluation metric** For object detection, mean average precision (i.e., mAP) is used for performance evaluation. On the VOC datasets, mAP is calculated under the IoU threshold of 0.5. On the COCO benchmark, mAP is calculated by averaged over the IoU thresholds of 0.5:0.95. For semantic segmentation, mean intersection over union (i.e., mIoU) is used for performance evaluation.

4.2. Ablation experiments on the VOC 2012 dataset

In this subsection, experiments are conducted on the PASCAL VOC 2012 to validate the effectiveness of proposed method. The set of VOC12-trainvalaug-seg is used for training and the set of VOC12-val-seg is for performance evaluation. The input images are rescaled to the size of $300 \times 300$, and the size of a mini-batch is 32. The total number of iteration in the training stage is 40k, and the initial learning rate is 0.0001. The learning rate decreases by a factor of 10 at the 25k and 35k iterations.

**PairNet** The top part of Table 1 shows the ablation experiments of PairNet. When the different layers of the decoder are only used for multi-scale object detection (i.e., Table 1(a)), mAP of object detection is 78.0%. When the fine layer of the decoder is used for semantic segmentation (i.e., Table 1(b)), mIoU of semantic segmentation is 72.5%. When all the different layers of the decoder are used for joint object detection and semantic segmentation (i.e., Table 1(d)), mAP and mIoU of PairNet are 78.9% and 73.1%. Namely, PairNet can improve the performance of both detection and segmentation, which indicates that joint detection and segmentation on each layer of the decoder is useful.

<table>
<thead>
<tr>
<th>Method</th>
<th>backbone</th>
<th>input size</th>
<th>det</th>
<th>seg_fine</th>
<th>seg_all</th>
<th>MFS</th>
<th>IC</th>
<th>CAS</th>
<th>ASF</th>
<th>mAP</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) only detection</td>
<td>ResNet50</td>
<td>300×300</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>78.0</td>
<td>N/A</td>
</tr>
<tr>
<td>(b) only segmentation with fine layer</td>
<td>ResNet50</td>
<td>300×300</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N/A</td>
<td>72.5</td>
</tr>
<tr>
<td>(c) only segmentation with all layers</td>
<td>ResNet50</td>
<td>300×300</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N/A</td>
<td>72.9</td>
</tr>
<tr>
<td>(d) PairNet</td>
<td>ResNet50</td>
<td>300×300</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>78.9</td>
<td>73.1</td>
</tr>
<tr>
<td>(e) add MFS</td>
<td>ResNet50</td>
<td>300×300</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>79.0</td>
<td>73.5</td>
</tr>
<tr>
<td>(f) add MFS and IC</td>
<td>ResNet50</td>
<td>300×300</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>79.5</td>
<td>73.6</td>
</tr>
<tr>
<td>(g) add MFS, IC, and CAS</td>
<td>ResNet50</td>
<td>300×300</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>79.7</td>
<td>74.4</td>
</tr>
<tr>
<td>(h) TripleNet</td>
<td>ResNet50</td>
<td>300×300</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>80.0</td>
<td>74.8</td>
</tr>
<tr>
<td>(i) TripleNet†</td>
<td>ResNet50</td>
<td>512×512</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>83.2</td>
<td>77.3</td>
</tr>
<tr>
<td>(j) TripleNet†</td>
<td>ResNet50</td>
<td>512×512</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>87.3</td>
<td>79.6</td>
</tr>
<tr>
<td>(k) TripleNet†</td>
<td>ResNet101</td>
<td>512×512</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>88.1</td>
<td>80.4</td>
</tr>
</tbody>
</table>

Table 1. Ablation experiments of PairNet and TripleNet on the VOC2012-val-seg set. The backbone is ResNet50 or ResNet 101 [16], and the input image is of $300 \times 300$ or $512 \times 512$. † means that VOC 2007 is use for training. “MFS” means multi-scale fused segmentation, “IC” means inner-connected module, “CAS” means class-agnostic segmentation, and “ASF” means attention skip-layer fusion.
Meanwhile, the method using all the different layers of the decoder for segmentation (i.e., Table 1(c)) has better performance than the method using only the last layer of the decoder for segmentation (i.e., Table 1(b)). The reason is that using all the layers of the decoder for semantic segmentation provides a deeper supervision for feature learning.

**TripleNet** The medium part of Table 1 shows the ablation experiments of TripleNet. Based on PairNet, TripleNet adds four modules (i.e., MFS, IC, CAS, and ASF). When adding the MFS module, TripleNet outperforms PairNet by 0.1% on detection and 0.4% on segmentation. When adding the MFS and IC modules, TripleNet outperforms PairNet by 0.6% on detection and 0.5% on segmentation. When adding all four modules, TripleNet has the best performance on both detection and performance, which outperforms the baselines (i.e., Table 1(a) and (b)) by 2.0% on object detection and 2.3% on semantic segmentation.

**Larger input and more data** The bottom part of Table 1 shows the affects of larger input size and more training data. it can be seen that with the larger input size (i.e., $512 \times 512$) and more training data (i.e., adding the VOC 2007 dataset), TripleNet can further improve performance of object detection and semantic segmentation. For example, TripleNet512 with more extra training data achieves 88.1% mAP on object detection and 80.4% mIoU on semantic segmentation.

**Qualitative results** Fig. 4 shows the detection and segmentation results of some methods in Table 1. The first two columns are the ground-truths of object detection and semantic segmentation. The results of only detection (i.e., “only det” in Table 1(a)) and only segmentation (i.e., “only seg” in Table 1(b)) are shown in the third and fourth columns. The results of PairNet in Table 1(d) and TripleNet in Table 1(h) are shown in fifth to eighth columns. In Fig. 4(a), the examples of detection and segmentation both improved by PairNet or TripleNet are given. For example, in the first row, “only det” and “only seg” both miss three potted plant, while PairNet only misses one potted plant and TripleNet does not miss any potted plant. In Fig. 4(b), the example of object detection improved is shown. “only detect” can only detect one ship, PairNet can detect three ships, and TripleNet can detect four ships. In Fig. 4(c), the example of semantic segmentation improved is shown. “only seg” recognizes the blue bag as a motorbike, while PairNet and TripleNet correctly classify the blue bag into the background.

**Feature visualization** Fig. 5 further visualizes the feature maps of the methods in Table 1 (i.e., “only det”, “only seg”, and TripleNet). On object detection and semantic seg-
Figure 5. Feature visualization of the methods in Table 1 (i.e., “only det”, “only seg”, and TripleNet). For detection, the first 9 feature maps of maximum response values where objects are detected are selected. For segmentation, the feature map of corresponding object category is shown.

Table 3. Detection results of some single-stage and state-of-the-art methods on the VOC 2007 test set without using COCO pre-training. All the methods are based on single scale test.

Table 2. TripleNet vs TripleNet-SS. TripleNet-ss means that scale-specific semantic segmentation is used in each layer, which is similar to detection.

Table 4. Results of object detection (mAP) and semantic segmentation (mIoU) on the VOC 2012 test set without using COCO pre-training. Inference time is tested on NVIDIA TitanX.

4.3. Comparison with state-of-the-art methods on the VOC 2007/2012 test dataset

In this section, TripleNet is compared with some state-of-the-art methods on the VOC 2007/2012 datasets [9]. Among these methods, SSD [31], RON [23], DSSD [10], DES [53], STDN [55], RefineDet [52], and DFPR [22] are only for object detection. FCN [34], ParseNet [32], DeepLab [5], DPN [33], RefineNet [26], PSPNet [54], DFN [50], and EncNet [51] are only for semantic segmentation. BlitzNet [8] is for joint detection and segmentation.

VOC 2007 The proposed TripleNet is firstly compared with these methods on the VOC 2007 dataset. The training data is VOC 2007 trainval set and VOC 2012 trainval set. The total number of iterations is 75k, and the initial learning rate is 0.0001. The learning rate decreases at the 45k and 60k iterations by a factor of 10. As only the ground-truth of object detection is provided, these methods are only evaluated on object detection. Table 3 shows mAP of these methods. TripleNet with the relatively small ResNet50 has already the same performance as DFPR512 [22] with the deep ResNet101. With the deep ResNet101, TripleNet has 83.0% mAP, which outperforms all the state-of-the-art methods.

VOC 2012 Then, the proposed method is compared with these methods on the VOC 2012 dataset. The training data is VOC 2007 dataset and VOC 2012 trainval set. The other parameter settings are the same as that used in the VOC 2007 dataset. Table 4 shows detection or segmentation results. It can be concluded as follows: (1) By using a single network, the proposed TripleNet achieves the state-of-the-art performance on both object detection and semantic seg-

unspecific semantic segmentation is adapted.


<table>
<thead>
<tr>
<th>Method</th>
<th>backbone</th>
<th>input size</th>
<th>$AP$</th>
<th>$AP_{50}$</th>
<th>$AP_{75}$</th>
<th>$AP_s$</th>
<th>$AP_m$</th>
<th>$AP_l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSD512 [31]</td>
<td>VGG16</td>
<td>512 × 512</td>
<td>28.8</td>
<td>48.5</td>
<td>30.3</td>
<td>10.9</td>
<td>31.8</td>
<td>43.5</td>
</tr>
<tr>
<td>DSSD512 [10]</td>
<td>ResNet101</td>
<td>512 × 512</td>
<td>33.2</td>
<td>53.3</td>
<td>35.2</td>
<td>13.0</td>
<td>35.4</td>
<td>51.1</td>
</tr>
<tr>
<td>STDN512 [55]</td>
<td>VGG16</td>
<td>512 × 512</td>
<td>31.8</td>
<td>51.0</td>
<td>33.6</td>
<td>14.4</td>
<td>36.1</td>
<td>43.4</td>
</tr>
<tr>
<td>DES512 [53]</td>
<td>VGG16</td>
<td>512 × 512</td>
<td>32.8</td>
<td>53.2</td>
<td>34.6</td>
<td>13.9</td>
<td>36.0</td>
<td>47.6</td>
</tr>
<tr>
<td>RetinaNet [28]</td>
<td>ResNet101</td>
<td>500 × 500</td>
<td>34.4</td>
<td>53.1</td>
<td>36.8</td>
<td>14.7</td>
<td>38.5</td>
<td>49.1</td>
</tr>
<tr>
<td>RefineDet512 [52]</td>
<td>ResNet101</td>
<td>512 × 512</td>
<td>36.4</td>
<td>57.5</td>
<td>39.5</td>
<td>16.6</td>
<td>39.9</td>
<td>51.4</td>
</tr>
<tr>
<td>RFBN512 [30]</td>
<td>VGG16</td>
<td>512 × 512</td>
<td>34.4</td>
<td>55.7</td>
<td>36.4</td>
<td>17.6</td>
<td>37.0</td>
<td>47.6</td>
</tr>
<tr>
<td>DFF512 [22]</td>
<td>ResNet101</td>
<td>512 × 512</td>
<td>34.6</td>
<td>54.3</td>
<td>37.3</td>
<td>14.7</td>
<td>38.1</td>
<td>51.9</td>
</tr>
<tr>
<td>TripleNet512</td>
<td>ResNet50</td>
<td>512 × 512</td>
<td>35.9</td>
<td>57.8</td>
<td>38.0</td>
<td>17.7</td>
<td>37.2</td>
<td>50.7</td>
</tr>
<tr>
<td>TripleNet512</td>
<td>ResNet101</td>
<td>512 × 512</td>
<td>37.4</td>
<td>59.3</td>
<td>39.6</td>
<td>18.5</td>
<td>39.0</td>
<td>52.7</td>
</tr>
</tbody>
</table>

Table 5. Detection results of single-stage methods on COCO test-dev set. All the methods are based on single-scale test.

![Joint performance & speed on the VOC 2012 test set.](image)

Figure 6. Joint performance & speed on the VOC 2012 test set.

<table>
<thead>
<tr>
<th>Method</th>
<th>backbone</th>
<th>$AP$</th>
<th>$AP_{50}$</th>
<th>$AP_{75}$</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlitzNet512 [8]</td>
<td>ResNet50</td>
<td>34.1</td>
<td>55.1</td>
<td>35.9</td>
<td>53.5</td>
</tr>
<tr>
<td>TripleNet512</td>
<td>ResNet50</td>
<td>36.0</td>
<td>57.7</td>
<td>37.8</td>
<td>58.3</td>
</tr>
<tr>
<td>TripleNet512</td>
<td>ResNet101</td>
<td>37.1</td>
<td>58.7</td>
<td>39.4</td>
<td>59.6</td>
</tr>
</tbody>
</table>

Table 6. Comparison of joint object detection and semantic segmentation on the COCO minival test.

In this paper, we proposed two fully convolutional networks (i.e., PairNet and TripleNet) for joint object detection and semantic segmentation. PairNet simultaneously predicts the objects of different scales by different layers and parses pixel semantic labels by all different layers. To further exploit the mutual benefits between detection and segmentation, TripleNet adds four modules (i.e., multi-scale fused segmentation, inner-connected module, class-agnostic segmentation supervision, and attention skip-layer fusion) to PairNet. Experiments demonstrate that TripleNet can achieve state-of-the-art performance on both object detection and semantic segmentation at a real-time speed.

5. Conclusion

In this paper, we proposed two fully convolutional networks (i.e., PairNet and TripleNet) for joint object detection and semantic segmentation. PairNet simultaneously predicts the objects of different scales by different layers and parses pixel semantic labels by all different layers. To further exploit the mutual benefits between detection and segmentation, TripleNet adds four modules (i.e., multi-scale fused segmentation, inner-connected module, class-agnostic segmentation supervision, and attention skip-layer fusion) to PairNet. Experiments demonstrate that TripleNet can achieve state-of-the-art performance on both object detection and semantic segmentation at a real-time speed.

6. Acknowledgments

It was supported by National Natural Science Foundation of China (No. 61632018), Postdoctoral Program for Innovative Talents (No. BX20180214), China Postdoctoral Science Foundation (No. 2018M641647), and the Nokia.


