

# Spatial Attention Pyramid Network for Unsupervised Domain Adaptation

Congcong Li<sup>1</sup>, Dawei Du<sup>2</sup>, Libo Zhang<sup>3,4</sup>, Longyin Wen<sup>5</sup>, Tiejian Luo<sup>1</sup>, Yanjun Wu<sup>3,4</sup>, and Pengfei Zhu

<sup>1</sup> University of Chinese Academy of Sciences, Beijing, China

<sup>2</sup> University at Albany, State University of New York, Albany, NY, USA

<sup>3</sup> Institute of Software Chinese Academy of Sciences, Beijing, China

<sup>4</sup> State Key Laboratory of Computer Science, ISCAS, Beijing, China

<sup>5</sup> JD Digits, Mountain View, CA, USA.

**Abstract.** Unsupervised domain adaptation is critical in various computer vision tasks, such as object detection, instance segmentation, and semantic segmentation, which aims to **alleviate performance degradation caused by domain-shift**. Most of previous methods rely on a single-mode distribution of source and target domains to align them with adversarial learning, leading to inferior results in various scenarios. To that end, in this paper, we design a new spatial attention pyramid network for unsupervised domain adaptation. Specifically, we first build the spatial pyramid representation to capture context information of objects at different scales. Guided by the task-specific information, **we combine the dense global structure representation and local texture patterns at each spatial location effectively using the spatial attention mechanism**. In this way, the network is enforced to focus on the discriminative regions with **context information for domain adaption**. We conduct extensive experiments on various challenging datasets for unsupervised domain adaptation on object detection, instance segmentation, and semantic segmentation, which demonstrates that our method performs favorably against the state-of-the-art methods by a large margin. Our source code is available at **code\_path**.

**Keywords:** Unsupervised domain adaptation, spatial attention pyramid, object detection, semantic segmentation, instance segmentation.

## 1 Introduction

Over the past few years, deep neural network (DNN) significantly pushes forward the state-of-the-art in several tasks in computer vision field, such as object detection [1,2], instance segmentation [3,4], and semantic segmentation [5,6]. Notably, the DNN-based methods rely on large-scale annotated training data, which is difficult to cover diverse application domains. That is, the feature distributions (*e.g.*, local texture, object appearance and global structure) between source domain and target domain are dissimilar or even completely different. To avoid expensive and time-consuming human annotation, unsupervised domain

adaptation is proposed to learn discriminative cross-domain representation in such domain shift circumstance [7].

Most of previous methods [8,9,10] attempt to globally align the entire distributions between the source and target domains. However, it is challenging to generate a unified adaptation function for various scene layouts and appearance variation of different objects. Recent methods focus on transferring texture and color statistics within object instances or local patches. To deal with domain adaptation in the object detection and instance segmentation tasks, the basic idea in [11,12] is to exploit discriminative features in bounding boxes of objects and attempt to align them across both source and target domains. However, the context information around the objects is not fully exploited, causing inferior results in some scenarios. Meanwhile, some domain adaptation methods for semantic segmentation [13,14] enforce the semantic consistency between the pixels or local patches of the two domains, leading to deficiencies of critical information from object-level patterns. To that end, recent methods [15,16] concatenate global context feature and instance-level feature for distribution alignment, and optimize the model based on several loss terms for global-level and local-level features with pre-set weights. However, this method fails to exploit the context information of objects, which is not optimal in challenging scenarios.

In this paper, we design the spatial attention pyramid network (SAPNet) to solve unsupervised domain adaptation for object detection, instance segmentation, and semantic segmentation. Inspired by spatial pyramid pooling [17], we construct the spatial pyramid representation with multi-scale feature maps, which integrates full of holistic (image-level) and local (regions of interest) semantic information. Meanwhile, we design a task-specific guided spatial attention mechanism, which learns to capture multi-scale context information. In this way, the discriminative semantic regions are attended in a soft manner to extract features for adversarial learning. Extensive experiments are conducted on various challenging domain-shift datasets, such as Cityscapes [18] to FoggyCityscapes [19], Cityscapes [18] to KITTI [20], PASCAL VOC [21] to Clipart [22], and GTA5 [23] to Cityscapes [18]. It is worth mentioning that the proposed method surpasses the state-of-the-art methods on various tasks, *i.e.*, object detection, instance segmentation, and semantic segmentation. For example, our SAPNet improves from Cityscapes [18] to FoggyCityscapes [19] by 3% mAP in terms of object detection, and achieves comparable accuracy from GTA5 [23] to Cityscapes [18] in terms of semantic segmentation.

**Contributions.** 1) We propose a new spatial attention pyramid network to solve the unsupervised domain adaptation task for object detection, instance segmentation, and semantic segmentation. 2) We develop a task-specific guided spatial attention pyramid learning strategy to merge multi-level semantic information in feature maps of the pyramid. 3) Extensive experiments conducted on various challenging domain-shift datasets for object detection, instance segmentation, and semantic segmentation, demonstrate the effectiveness of the proposed method, surpassing the state-of-the-art methods.

## 2 Related Works

**Unsupervised domain adaptation.** Several methods have been proposed for unsupervised domain adaptation in terms of several tasks such as object detection [11,15,16], instance segmentation [12] and semantic segmentation [8,13,14]. For object detection domain adaptation, Chen *et al.* [11] align source and target domain both on image level and instance level using the gradient reverse layer [24]. Zhu *et al.* [12] mine the discriminative regions (pertinent to object detection) using  $k$ -means clustering and align them across both domains, which is applied in object detection and instance segmentation. Recently, the strong-weak adaptation method is proposed in [15]. It focuses the adversarial alignment loss toward images that are globally similar, and away from images that are globally dissimilar by employing focal loss [25]. Shen *et al.* [16] propose a gradient detach based stacked complementary losses method that adapt source domain and target domain on multiple layers. On the other hand, Hoffman *et al.* [8] perform global domain alignment in a novel semantic segmentation network with fully convolutional domain adversarial learning. Tsai *et al.* [13] learn discriminative feature representations of patches in the source domain by discovering multiple modes of patch-wise output distribution through the construction of a clustered space. Luo *et al.* [14] introduce a category-level adversarial network to enforce local semantic consistency on the output space using two distinct classifiers.

The aforementioned methods have achieved remarkable improvements compared with the non-adapted method by considering scene layouts and texture/color statistics in local regions. However, they only consider domain adaptation in two levels, *i.e.*, aligning the feature maps of the whole image or local regions with a fixed scale. Different from them, we design the spatial pyramid representation to capture multi-level semantic patterns within the image for better adaptation between the source domain and target domain.

**Attention mechanism.** To focus on the most discriminative features in the network, various attention mechanisms have been explored. SENet [26] develops the Squeeze-and-Excitation (SE) block that adaptively recalibrates channel-wise feature responses by explicitly modeling interdependencies between channels. Non-local networks [27] capture long-range dependencies by computing the response at a position as a weighted sum of the features at all positions in the input feature maps. SKNet [28] uses softmax attention to weighted fuse multiple feature maps of different kernel sizes to adaptively adjust the receptive field size of the input feature map. Except channel-wise attention, CBAM [29] introduce spatial attention by calculating the inter-spatial relation of features. To highlight transferable regions in domain adaptation, Wang *et al.* [30] use multiple region-level domain discriminators and single image-level domain discriminator to generate transferable local and global attention, respectively. Sun and Wu [31] integrates atrous spatial pyramid, cascade attention mechanism and residual connections for image synthesis and image-to-image translation.

As previous works [32,33] have shown the importance of multi-scale information, we propose the attention pyramid learning to better adapt the source domain and target domain. Specifically, we employ the task-specific information

to guide pyramid attention to make full use of semantic information of different feature maps at different levels.

**Detection and segmentation networks.** The performance of object detection and segmentation has boosted with the development of deep convolutional neural networks. Faster R-CNN [1] is a classic object detection framework that predicts class-agnostic coarse object proposals using a region proposal network (RPN) and then extract fix-sized object features to classify object category and refine object location. Moreover, He *et al.* [3] extend Faster R-CNN by adding a branch for predicting instance segmentation results. For semantic segmentation, the DeepLab-v2 method [34] develops Atrous spatial pyramid pooling (ASPP) modules to segment objects at multiple scales. For fair comparison, this work proposes the spatial attention pyramid network based on the same detection and segmentation frameworks as that in the previous domain adaptation methods.

### 3 Spatial Attention Pyramid Network

We design a spatial attention pyramid network (SAPNet) to solve various computer vision tasks, such as object detection, instance segmentation, and semantic segmentation. First of all, we define the labeled source domain  $\mathcal{D}_s$  and the unlabeled target domain  $\mathcal{D}_t$ , which are subject to the complex and unknown distributions in the source and target domains. Our goal is to find discriminative representation for the distributions of both source and target domains to capture various semantic information and local patterns in different tasks. The architecture of SAPNet is presented in Fig. 1.

**Spatial pyramid representation.** According to [17,35], spatial pyramid pooling can maintain spatial information by pooling in local spatial bins. To better adapt source and target domains, we develop a spatial pyramid representation to exploit the underlying distribution within an image.

Specifically, as shown in Fig. 1, the feature map  $\hat{f} \in \mathbb{R}^{\hat{C} \times H \times W}$  is extracted from the backbone  $G$  of the network, where  $\hat{C}$ ,  $H$  and  $W$  are the channel dimension, height, width of feature maps respectively. To improve efficiency, we first reduce the number of channels in  $\hat{f}$  to  $\bar{f} \in \mathbb{R}^{C \times H \times W}$  gradually by using three  $1 \times 1$  convolutional layers, *i.e.*, we set  $C = 256$  in all our experiments. Second, we use multiple average pooling layers with different sizes to operate upon the feature map  $\bar{f}$  separately. The sizes of the pooling operation are  $\{k^n\}_{n=1}^N$ , where  $N$  is the number of pooling layers. That is, the rectangular pooling region with the size  $k^n$  at each location of  $\bar{f}$  is down-sampled to the average value of each region, resulting in the pyramid of  $N$  pooled feature maps  $\{f^1, \dots, f^N\}$ . In this way, every pooled feature map  $f^n \in \mathbb{R}^{C \times H_n \times W_n}$  in the pyramid can encode different semantic information of objects or layouts within the image.

It is worth mentioning that the proposed spatial pyramid representation is related with the spatial pyramid pooling (SPP) module for visual recognition [17]. While they share the pooling concept, we would like to highlight two important differences as follows.

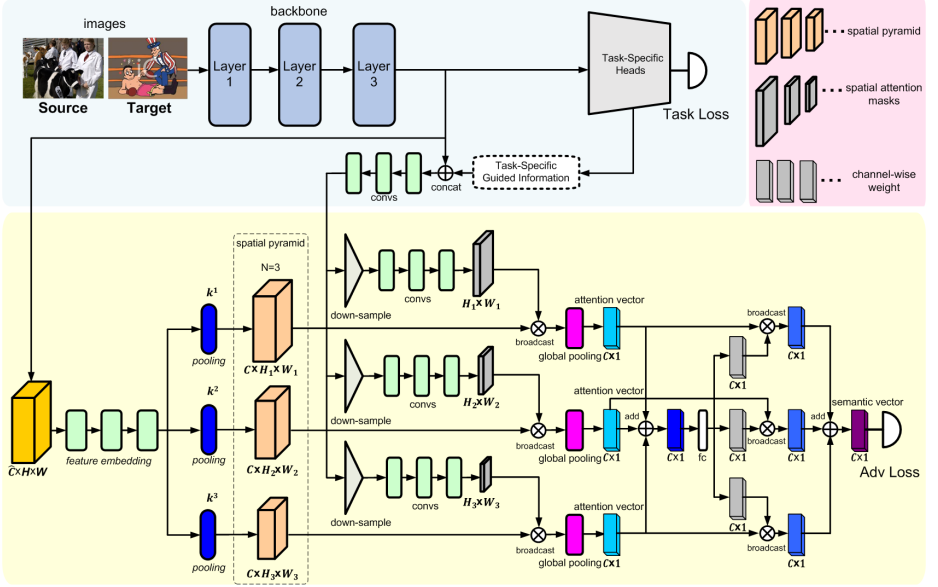


Fig. 1: The framework of spatial attention pyramid network. For clarity, we only show  $N = 3$  levels in the spatial pyramid.

- First, we use average pooling instead of max pooling to construct the spatial pyramid representation. It can better capture the overall strength of local patterns such as edges and textures, which is demonstrated in the ablation study.
- Second, the SPP module concatenates all the outputs of spatial pyramid pooling to a fixed-dimensional vector and feeds it into the following fully-connected layer. It is difficult for SPP module to make full use of the deep spatial pyramid, due to the huge computational complexity of the concatenated high-dimensional vector from a number of feature maps in the pyramid.

**Attention mechanism.** Moreover, we integrate the spatial attention pyramid strategy to enforce the network to focus on the most discriminative semantic regions and feature maps. There are mainly two advantages of introducing the attention mechanism in the spatial pyramid architecture. First, there exist different local patterns in each spatial location of feature maps. Second, different feature maps in the pyramid have different contributions to the semantic representation. The detailed learning method is described in two aspects as follows.

To facilitate highlighting the most discriminative semantic regions, the spatial attention masks for the pyramid  $\{f^1, \dots, f^N\}$  are learned based on the guided information from the task-specific heads (*i.e.*, object detection, instance segmentation and semantic segmentation). For object detection and instance segmentation, the guided information corresponds to the output map with the

size of  $A \times H \times W$  from the classification head of region proposal network (RPN). It can predict object's confidence in terms of the locations in feature maps for all the number of anchors  $A$ . For semantic segmentation, the guided information is the output map with the size of  $C_{sem} \times H \times W$  from the segmentation head, where  $C_{sem}$  is the number of semantic categories. We denote the guided map as  $\hat{\mathcal{P}}$ .

Then, we concatenate the guided map  $\hat{\mathcal{P}}$  and feature map  $\hat{f}$  to generate guided feature map  $\bar{\mathcal{P}}$  followed by 3 convolutional layers. The guided feature map  $\bar{\mathcal{P}}$  is shared for all  $N$  scales. To adjust each scale of feature map  $f^n$  in the spatial pyramid, we resize  $\bar{\mathcal{P}}$  to the size  $C \times H_n \times W_n$  at each level. The spatial attention mask  $\mathcal{P}^n \in \mathbb{R}^{H_n \times W_n}$  can be predicted by the followed 3 convolutional layers. Finally,  $\mathcal{P}^n$  is normalized using the softmax function to compute the spatial attention, *i.e.*,

$$\omega^n(x, y) = \frac{e^{\mathcal{P}^n(x, y)}}{\sum_{i=1}^{H_n} \sum_{j=1}^{W_n} e^{\mathcal{P}^n(i, j)}}, \quad (1)$$

where  $\omega^n(i, j)$  indicates the value of the attention mask  $\omega^n$  at  $(i, j)$ . Thus, we have  $\sum_{i=1}^{H_n} \sum_{j=1}^{W_n} \omega^n(i, j) = 1$ .

As shown in Fig. 2, we provide some examples with normalized attention masks  $\omega^n$  for different feature maps in the spatial pyramid when  $N = 7$ . We can conclude that the feature map with different scale  $k^n$  focuses on different semantic regions. For example, in the forth row, the feature map with smaller pooling size ( $k = 3$ ) pays more attention on the seagull, while the feature map with larger pooling size ( $k = 21$ ) focuses on the sail boat and the neighbouring context. Based on different guided information,  $\omega^n$  recalibrates spatial responses in feature map  $f^n$  adaptively.

On the other hand, it can be seen that not all the attention masks correspond to meaningful regions (see the attention mask with pooling size  $k = 37$  in the forth row). Inspired by [28], we develop a dynamic weight selection mechanism to adjust the channel-wise weight of feature maps in the pyramid adaptively. To consider feature maps with different size, we normalize  $f^n$  to an attention vector  $V^n \in \mathbb{R}^{C \times 1}$  using the corresponding spatial attention weight  $\omega^n$  as:

$$V^n = \sum_{i=1}^{H_n} \sum_{j=1}^{W_n} f^n \cdot \omega^n, \quad (2)$$

where  $i, j$  enumerate all spatial positions of weighted feature map  $f^n \cdot \omega^n$ . Thus the attention vectors have the same size for all the feature maps in the pyramid. Given attention vectors  $\{V^1, \dots, V^N\}$ , we first fuse these vectors via an element-wise addition, *i.e.*,  $v = \sum_{n=1}^N V^n$ . Then, a compact feature  $\mathbf{z} \in \mathbb{R}^{d \times 1}$  is created to enable the guidance for adaptive selections by the batch normalization layer. After that, for each attention vector  $V^n$ , we compute the channel-wise attention weight  $\phi^n \in \mathbb{R}^{C \times 1}$  as

$$\phi^n = \frac{e^{a_n \cdot \mathbf{z}}}{\sum_{i=1}^N e^{a_i \cdot \mathbf{z}}}, \quad (3)$$

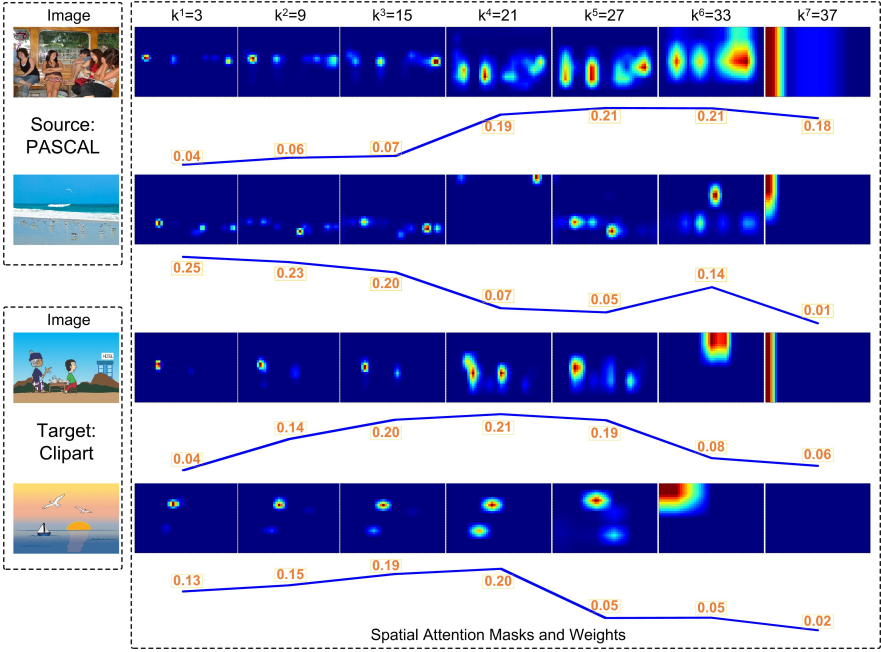


Fig. 2: Visualization of spatial attention masks and corresponding weights (blue lines) for the spatial pyramid. The attention masks of different feature maps are resized by the same scale for better visualization. Note the differences of different scale spatial attention mask responses and how SAPNet adaptively recalibrates their weights when the object sizes and scene structure change.

where  $\{a_i, \dots, a_N\}$  are learnable parameters of fully connected layers for each scale. We have  $\sum_{c=1}^N \phi^n(c) = 1$ , where  $\phi^n(c)$  is the  $c$ -th element of  $\phi^n$ . As shown in Fig. 2, we show the corresponding weights of each feature map in the spatial pyramid. Specifically, for each image, we compute the mean of channel-wise attention weight  $\phi^n$  for each scale. Finally, the fused semantic vector  $\mathcal{V} \in \mathbb{R}^{C \times 1}$  is obtained through the channel-wise attention weight as

$$\mathcal{V} = \sum_{n=1}^N V^n \cdot \phi^n, \quad (4)$$

where  $\mathcal{V}$  is a highly embedded vector in the latent space that encodes the semantic information of different spatial locations, different channels and the relations among them.

**Optimization.** The whole network is trained by minimizing two loss terms, *i.e.*, adversarial loss and task-specific loss. The adversarial loss is used to determine whether the sample comes from the source domain or target domain. Specifically, we calculate the probability  $x_i$  of the sample belonging to the target domain



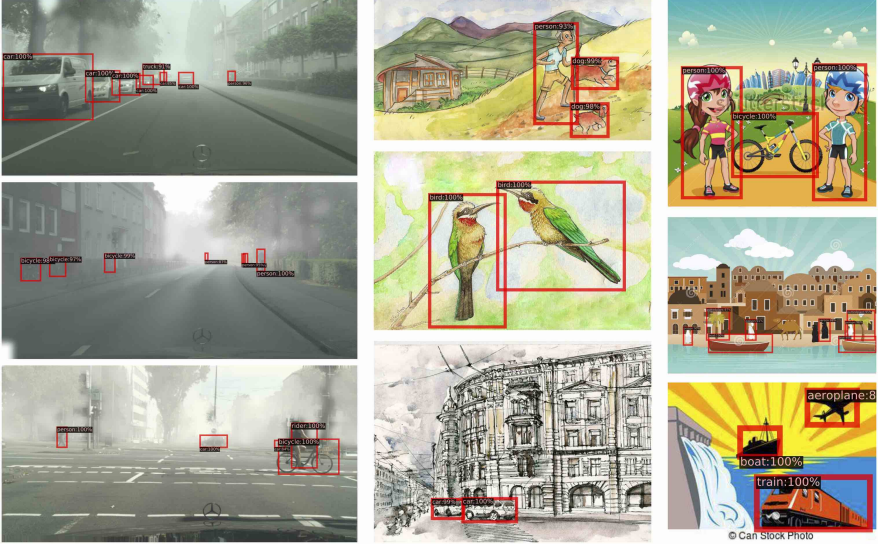


Fig. 3: Adaptation object detection results. From left to right: Foggy Cityscapes, Watercolor and Clipart.

based on the fused semantic vector  $\mathcal{V}$  using a simple fully-connected layer. The proposed SAPNet is denoted as  $D$ . Then, the adversarial loss is computed as

$$\mathcal{L}_{\text{adv}}(G, D) = \frac{1}{|\mathcal{D}_s \cup \mathcal{D}_t|} \sum_{x_i \in \mathcal{D}_s \cup \mathcal{D}_t} \mathcal{H}(D(G(x_i)), y_i), \quad (5)$$

where  $y_i$  is the domain label (0 for source domain and 1 for target domain) and  $\mathcal{H}$  is the cross-entropy loss function. On the other hand, the task loss  $\mathcal{L}_{\text{task}} = \{\mathcal{L}_{\text{det}}, \mathcal{L}_{\text{ins}}, \mathcal{L}_{\text{seg}}\}$  is determined by the specific task, *i.e.*, object detection, instance segmentation, and semantic segmentation. The loss is computed as

$$\mathcal{L}_{\text{task}}(G, R) = \frac{1}{|\mathcal{D}_s|} \sum_{x_i \in \mathcal{D}_s} \mathcal{L}_{\text{task}}(R(G(x_i)), y_i^s), \quad (6)$$

where  $G$  and  $R$  are the backbone and task-specific components of the network, respectively.  $y_i^s$  is the ground-truth label of sample  $i$  in the source domain. Taking object detection as an example, we denote the objective of Faster R-CNN as  $\mathcal{L}_{\text{det}}$ , which contains classification loss of object categories and regression loss of object bounding boxes.

In summary, the overall objective is formulated as

$$\max_D \min_{G, R} \mathcal{L}_{\text{task}}(G, R) - \lambda \mathcal{L}_{\text{adv}}(G, D), \quad (7)$$



Table 1: Adaptation detection results from Cityscapes to FoggyCityscapes.

| Method               | person      | rider       | car         | truck       | bus         | train       | cycle       | bicycle     | mAP         |
|----------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Faster R-CNN (w/o)   | 24.1        | 33.1        | 34.3        | 4.1         | 22.3        | 3.0         | 15.3        | 26.5        | 20.3        |
| DA-Faster [11]       | 25.0        | 31.0        | 40.5        | 22.1        | 35.3        | 20.2        | 20.0        | 27.1        | 27.6        |
| SCDA [12]            | 33.5        | 38.0        | 48.5        | 26.5        | 39.0        | 23.3        | 28.0        | 33.6        | 33.8        |
| Strong-Weak [15]     | 29.9        | 42.3        | 43.5        | 24.5        | 36.2        | 32.6        | 30.0        | 35.3        | 34.3        |
| Diversify&match [37] | 30.8        | 40.5        | 44.3        | 27.2        | 38.4        | 34.5        | 28.4        | 32.2        | 34.6        |
| MAF [38]             | 28.2        | 39.5        | 43.9        | 23.8        | 39.9        | 33.3        | 29.2        | 33.9        | 34.0        |
| SCL [16]             | 31.6        | 44.0        | 44.8        | <b>30.4</b> | 41.8        | <b>40.7</b> | <b>33.6</b> | 36.2        | 37.9        |
| SAPNet               | <b>40.8</b> | <b>46.7</b> | <b>59.8</b> | 24.3        | <b>46.8</b> | 37.5        | 30.4        | <b>40.7</b> | <b>40.9</b> |

where  $\lambda$  controls the trade-off between task-specific loss and adversarial training loss. Following [11, 15], we use gradient reverse layer (GRL) [24] to enable adversarial training where the gradient is reversed before back-propagating to  $G$  from  $D$ . We first train the networks with only source domain to avoid initially noisy predictions. Then we train the whole model with Adam optimizer and the initial learning rate is set to  $10^{-5}$ , then divided by 10 at 70,000, 80,000 iterations. The total number of training iterations is 90,000.

## 4 Experiment

We implement our SAPNet method with PyTorch [36], which is evaluated in three domain adaptation tasks, including object detection, instance segmentation, and semantic segmentation. For fair comparison, we set the shorter side of the image to 600 following the implementation of [15, 16] with RoIAlign [3] in object detection; for instance segmentation and semantic segmentation, we use the same settings as previous methods. To consider the trade-off between accuracy and complexity, the number of pyramid levels is set to  $N = 13$  for object detection and instance segmentation, *i.e.*, we have the spatial pooling size set  $K = \{3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 35, 37\}$ . Note that we start from the initial pooling size  $3 \times 3$  with the step of 3, and the last two pooling sizes are reduced from  $\{38, 41\}$  to  $\{35, 37\}$  because of the width limit of feature map. For semantic segmentation, the number of pyramid levels is set to  $N = 9$  since semantic segmentation involves feature maps with higher resolution, *i.e.*,  $K = \{3, 9, 15, 21, 27, 33, 39, 45, 51\}$ .

### 4.1 Domain Adaptation for Detection

For object detection task, we conduct our experiments in three different domain shift scenarios: (1) similar domains; (2) dissimilar domains; and (3) from synthetic to real images. We compare our model to the state-of-the-arts on 6 domain shift datasets: Cityscapes [18] to FoggyCityscapes [19], Cityscapes [18] to KITTI [20], KITTI [20] to Cityscapes [18], PASCAL VOC [21] to Clipart [22], PASCAL

Table 2: Adaptation detection results between KITTI and Cityscapes. We report AP scores in terms of the car category on both directions, including KITTI  $\rightarrow$  Cityscapes and Cityscapes  $\rightarrow$  KITTI.

| Method                      | KITTI $\rightarrow$ Cityscapes | Cityscapes $\rightarrow$ KITTI |
|-----------------------------|--------------------------------|--------------------------------|
| Faster RCNN                 | 30.2                           | 53.5                           |
| DA-Faster [11]              | 38.5                           | 64.1                           |
| Strong-Weak (impl. of [16]) | 37.9                           | 71.0                           |
| SCL [16]                    | 41.9                           | 72.7                           |
| SCDA [12]                   | 42.5                           | -                              |
| SAPNet                      | <b>43.4</b>                    | <b>75.2</b>                    |

VOC [21] to Watercolor [22], Sim10K [39] to Cityscapes [18]. For fair comparison, we use ResNet101 and VGG-16 as the backbone and the last convolutional layer to enable domain adaptation as similar as that in [15,16]. Some qualitative adaptation results of object detection are shown in Fig. 3.

**Cityscapes  $\rightarrow$  FoggyCityscapes.** To verify our method in the similar domain adaptation scenarios, we evaluate our model between Cityscapes [18] and FoggyCityscapes [19] (simulated attenuation coefficient  $\beta = 0.02$ ) at the most difficult level. Specifically, Cityscapes is the source domain, while the target domain FoggyCityscape (Foggy for short) is rendered from the same images in Cityscape using depth information. We set  $\lambda = 1.0$  in (7) empirically. As shown in Table 1, our SAPNet gains 3.0% average accuracy improvement compared with the previous state-of-the-art methods. Specifically, in terms of *person* and *car* categories, our method outperforms the second performer with a huge margin (about 9% and 15% higher, respectively).

**Cityscapes  $\leftrightarrow$  KITTI.** As shown in Table 2, we present the comparison between our model and state-of-the-arts on domain adaptation between Cityscapes [18] and KITTI [20]. Similar to the works in [11,16], we use KITTI training set that contains 7,481 images. We set  $\lambda = 0.01$  for Cityscapes  $\rightarrow$  KITTI and  $\lambda = 0.2$  for KITTI  $\rightarrow$  Cityscapes in (7) empirically. The Strong-Weak and SCL [16] methods only employ multi-stage feature maps from the backbone to align holistic features, resulting in inferior performance than our method on both directions. In summary, our method achieves 1.5% and 2.5% accuracy improvement of KITTI  $\rightarrow$  Cityscapes and Cityscapes  $\rightarrow$  KITTI, respectively.

**PASCAL VOC  $\rightarrow$  Clipart/WaterColor.** Moreover, we evaluate our method on dissimilar domains, *i.e.*, from real images to artistic images. According to [15], PASCAL VOC [21] is the source domain, where the PASCAL VOC 2007 and 2012 training and validation sets are used for training. For the target domain, we use Clipart [22] and Watercolor [22] as that in [15]. ResNet-101 [40] pre-trained on ImageNet [41] is used as the backbone network following [15,16]. We set  $\lambda = 0.1$  and  $\lambda = 0.01$  for Clipart [22] and Watercolor [22] respectively. As shown in Table 3 and Table 4, our model obtains comparable results with SCL [16].

Table 3: Adaptation detection results from PASCAL VOC to Clipart.

| Method           | aero        | bicycle     | bird        | boat        | bottle      | bus         | car         | cat         | chair       | cow         | mAP |
|------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-----|
| FRCNN [1]        | 35.6        | 52.5        | 24.3        | 23.0        | 20.0        | 43.9        | 32.8        | 10.7        | 30.6        | 11.7        |     |
| BDC-Faster [15]  | 20.2        | 46.4        | 20.4        | 19.3        | 18.7        | 41.3        | 26.5        | 6.4         | 33.2        | 11.7        |     |
| DA-Faster [11]   | 15.0        | 34.6        | 12.4        | 11.9        | 19.8        | 21.1        | 23.2        | 3.1         | 22.1        | 26.3        |     |
| WST-BSR [42]     | 28.0        | 64.5        | 23.9        | 19.0        | 21.9        | <b>64.3</b> | 43.5        | 16.4        | 42.2        | 25.9        |     |
| Strong-Weak [15] | 26.2        | 48.5        | 32.6        | <b>33.7</b> | 38.5        | 54.3        | 37.1        | 18.6        | 34.8        | 58.3        |     |
| SCL [16]         | <b>44.7</b> | 50.0        | <b>33.6</b> | 27.4        | 42.2        | 55.6        | 38.3        | <b>19.2</b> | 37.9        | <b>69.0</b> |     |
| Ours             | 27.4        | <b>70.8</b> | 32.0        | 27.9        | <b>42.4</b> | 63.5        | <b>47.5</b> | 14.3        | <b>48.2</b> | 46.1        |     |

| Method           | table       | dog         | horse       | bike        | person      | plant       | sheep       | sofa        | train       | tv          | mAP         |
|------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| FRCNN [1]        | 13.8        | 6.0         | 36.8        | 45.9        | 48.7        | 41.9        | 16.5        | 7.3         | 22.9        | 32.0        | 27.8        |
| BDC-Faster [15]  | 26.0        | 1.7         | 36.6        | 41.5        | 37.7        | 44.5        | 10.6        | 20.4        | 33.3        | 15.5        | 25.6        |
| DA-Faster [11]   | 10.6        | 10.0        | 19.6        | 39.4        | 34.6        | 29.3        | 1.0         | 17.1        | 19.7        | 24.8        | 19.8        |
| WST-BSR [42]     | 30.5        | 7.9         | 25.5        | 67.6        | 54.5        | 36.4        | 10.3        | 31.2        | <b>57.4</b> | 43.5        | 35.7        |
| Strong-Weak [15] | 17.0        | 12.5        | 33.8        | 65.5        | 61.6        | 52.0        | 9.3         | 24.9        | 54.1        | 49.1        | 38.1        |
| SCL [16]         | 30.1        | <b>26.3</b> | 34.4        | 67.3        | 61.0        | 47.9        | <b>21.4</b> | <b>26.3</b> | 50.1        | 47.3        | 41.5        |
| Ours             | <b>31.8</b> | 17.9        | <b>43.8</b> | <b>68.0</b> | <b>68.1</b> | <b>49.0</b> | 18.7        | 20.4        | 55.8        | <b>51.3</b> | <b>42.2</b> |

Table 4: Adaptation detection results from PASCAL VOC to WaterColor.

| Method           | bike        | bird        | car         | cat         | dog         | person      | mAP         |
|------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Faster RCNN      | 68.8        | 46.8        | 37.2        | 32.7        | 21.3        | 60.7        | 44.6        |
| DA-Faster [11]   | 75.2        | 40.6        | 48.0        | 31.5        | 20.6        | 60.0        | 46.0        |
| Strong-Weak [15] | <b>82.3</b> | <b>55.9</b> | 46.5        | 32.7        | 35.5        | 66.7        | 53.3        |
| SCL [16]         | 82.2        | 55.1        | 51.8        | <b>39.6</b> | 38.4        | 64.0        | <b>55.2</b> |
| Ours             | 81.1        | 51.1        | <b>53.6</b> | 34.3        | <b>39.8</b> | <b>71.3</b> | <b>55.2</b> |

**Sim10K→Cityscapes.** In addition, we evaluate our model in the synthetic to real scenario. Following [11,15], we use Sim10K [39] as the source domain that contains 10,000 training images collected from the computer game Grand Theft Auto 5 (GTA5). We set  $\lambda = 0.1$  in (7) empirically. As shown in Table 5, our SAPNet obtains 3.3% improvement in terms of AP score compared with state-of-the-art methods.

It is worth mentioning that BDC-Faster [15] is also trained using cross-entropy loss but the performance is significantly decreased. Therefore, The Strong-Weak method [15] adapts the focal loss [25] to balance different regions. Compared with [15,16], our proposed attention mechanism is much more effective and thus the focal loss module is no longer needed.

## 4.2 Domain Adaptation for Segmentation

**Instance Segmentation.** For instance segmentation task, we evaluate our model from Cityscapes [18] to FoggyCityscapes [19]. Similar to [12], we use the VGG16 as the backbone network and add the segmentation head similar to that in Mask R-CNN [3]. From Table 6. We can conclude that our method outperforms SCDA [12] significantly, *i.e.*, 39.4 vs. 31.4. Some visual examples of adaptation instance segmentation results are shown in Fig. 4.

**Semantic Segmentation.** For semantic segmentation task, we conduct experiments from GTA5 [23] to Cityscapes [18] and SYNTHIA [43] to Cityscapes.

Table 5: Adaptation detection results from Sim10k to Cityscapes.

| Method           | AP on Car   |
|------------------|-------------|
| Faster R-CNN     | 34.6        |
| DA-Faster [11]   | 38.9        |
| Strong-Weak [15] | 40.1        |
| SCL [16]         | 42.6        |
| SCDA [12]        | 43.0        |
| Ours             | <b>44.9</b> |

Table 6: Adaptation instance segmentation results from Cityscapes to FoggyCityscapes.

| Method      | mAP         |
|-------------|-------------|
| Source Only | 26.6        |
| SCDA [12]   | 31.4        |
| Ours        | <b>39.4</b> |

Fig. 4: Instance segmentation results for Cityscapes  $\rightarrow$  Foggy Cityscapes.

Following [14], we use the DeepLab-v2 [34] framework with ResNet-101 backbone that is pre-trained on ImageNet. Notably, the task-specific guided map for semantic segmentation naturally comes from the predicted output with the shape of  $C_{sem} \times H \times W$ , where  $C_{sem}$  is the number of semantic categories. As presented in Table 7 and Table 8, our method achieves comparable segmentation accuracy with state-of-the-arts on the domain adaptation from GTA5 [23] to Cityscapes [18], and from SYNTHIA [43] to Cityscapes. Some visual examples of adaptation semantic segmentation results are shown in Fig. 5.

### 4.3 Ablation Study

We further perform experiments to study the effect of important aspects in SAP-Net, *i.e.*, task-specific guided map and spatial attention pyramid. Since PASCAL VOC  $\rightarrow$  Clipart, Sim10k  $\rightarrow$  Cityscapes and Cityscapes  $\rightarrow$  Foggy represent three different domain shift scenarios (*i.e.*, similar domains, dissimilar domains, and from synthetic to real images), we perform ablation study in terms of object detection datasets for comprehensive analysis.

**Effectiveness of task-specific guided information.** To investigate the importance of task-specific guided information, we remove the task-specific guid-

Table 7: Adaptation semantic segmentation results from GTA5 to Cityscapes.

| Method    | road        | side        | buil.       | wall        | fence       | pole        | light       | sign        | vege.       | terr.       |
|-----------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Source    | 75.8        | 16.8        | 77.2        | 12.5        | 21.0        | 25.5        | 30.1        | 20.1        | 81.3        | 24.6        |
| ROAD [44] | 76.3        | 36.1        | 69.6        | 28.6        | 22.4        | <b>28.6</b> | 29.3        | 14.8        | 82.3        | <b>35.3</b> |
| TAN [45]  | 86.5        | 25.9        | <b>79.8</b> | 22.1        | 20.0        | 23.6        | 33.1        | 21.8        | 81.8        | 25.9        |
| CLAN [14] | 87.0        | 27.1        | 79.6        | 27.3        | 23.3        | 28.3        | <b>35.5</b> | <b>24.2</b> | <b>83.6</b> | 27.4        |
| Ours      | <b>88.4</b> | <b>38.7</b> | 79.5        | <b>29.4</b> | <b>24.7</b> | 27.3        | 32.6        | 20.4        | 82.2        | 32.9        |

---

|           | sky         | pers.       | rider       | car         | truck       | bus         | train      | motor       | bike        | mIoU        |
|-----------|-------------|-------------|-------------|-------------|-------------|-------------|------------|-------------|-------------|-------------|
| Source    | 70.3        | 53.8        | 26.4        | 49.9        | 17.2        | 25.9        | 6.5        | 25.3        | <b>36.0</b> | 36.6        |
| ROAD [44] | 72.9        | 54.4        | 17.8        | 78.9        | 27.7        | 30.3        | 4.0        | 24.9        | 12.6        | 39.4        |
| TAN [45]  | <b>75.9</b> | 57.3        | 26.2        | 76.3        | 29.8        | 32.1        | <b>7.2</b> | 29.5        | 32.5        | 41.4        |
| CLAN [14] | 74.2        | <b>58.6</b> | <b>28.0</b> | 76.2        | <b>33.1</b> | 36.7        | 6.7        | <b>31.9</b> | 31.4        | 43.2        |
| Ours      | 73.3        | 55.5        | 26.9        | <b>82.4</b> | 31.8        | <b>41.8</b> | 2.4        | 26.5        | 24.1        | <b>43.2</b> |

Table 8: Adaptation results from SYNTHIA to Cityscapes.

| Method    | road | side | buil. | light | sign | vege. | sky  | pers. | rider | car  | bus  | motor | bike | mIoU |
|-----------|------|------|-------|-------|------|-------|------|-------|-------|------|------|-------|------|------|
| Source    | 55.6 | 23.8 | 74.6  | 6.1   | 12.1 | 74.8  | 79.0 | 55.3  | 19.1  | 39.6 | 23.3 | 13.7  | 25.0 | 38.6 |
| TAN [45]  | 79.2 | 37.2 | 78.8  | 9.9   | 10.5 | 78.2  | 80.5 | 53.5  | 19.6  | 67.0 | 29.5 | 21.6  | 31.3 | 45.9 |
| CLAN [14] | 81.3 | 37.0 | 80.1  | 16.1  | 13.7 | 78.2  | 81.5 | 53.4  | 21.2  | 73.0 | 32.9 | 22.6  | 30.7 | 47.8 |
| Ours      | 81.7 | 33.5 | 75.9  | 7.0   | 6.3  | 74.8  | 78.9 | 52.1  | 21.3  | 75.7 | 30.6 | 10.8  | 28.0 | 44.3 |

ance to generate the spatial attention mask, which is denoted as “w/o GM”. In this way, the number of channels of the first convolutional layer after concatenation of feature maps in layer 3 and task-specific guided information is reduced (see Fig. 1). However, the impact is negligible since the channel number of guided map is small. As presented in Table 9, the task-specific guided information improves the accuracy, especially for dissimilar domains PASCAL VOC and Clipart (42.2 vs. 37.1). We speculate that such guidance can facilitate focusing on the most discriminative semantic regions for domain adaptation.

**Effectiveness of spatial attention pyramid.** To investigate the effectiveness of spatial attention pyramid, we construct the “w/o SA” variant of SAPNet, which indicates that we remove the spatial attention masks and global attention pyramid (since no multi-scale vectors are available) in Fig. 1. As shown in Table 9, the performance drops dramatically without spatial attention pyramid. On the other hand, along with the increasing number of pooled feature maps in the pyramid, the performance is gradually improved. Specifically, we use the spatial pooling size set  $K = \{3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 35, 37\}$  when  $N = 13$ ,  $K = \{3, 9, 15, 21, 27, 33, 37\}$  when  $N = 7$  and  $K = \{3, 21, 37\}$  when  $N = 3$ . It indicates that the spatial pyramid with deep levels contains more discriminative semantic information for domain adaptation and our method can make full use of it. In addition, we compare average pooling and maximal pooling operations in spatial attention pyramid. We can conclude that average pooling achieves better performance in different datasets, which demonstrates the effectiveness of average pooling to capture discriminative local patterns for domain adaptation.

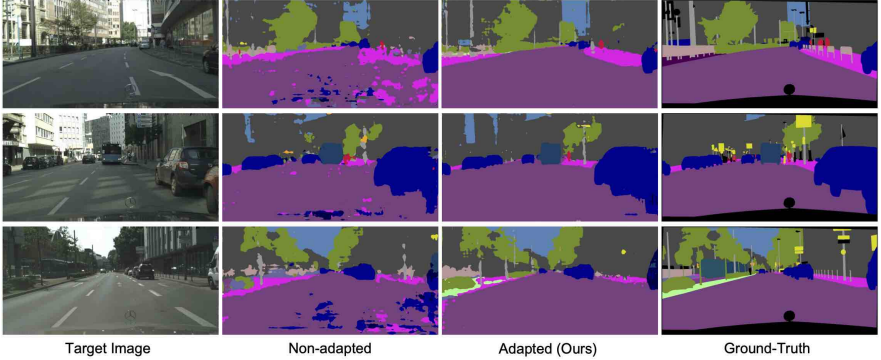
Fig. 5: Semantic segmentation results for GTA5  $\rightarrow$  Cityscapes.

Table 9: Effectiveness of important aspects in SAPNet.

| Variant         | PASCAL VOC $\rightarrow$ Clipart | Sim10k $\rightarrow$ Cityscapes | Cityscapes $\rightarrow$ Foggy |
|-----------------|----------------------------------|---------------------------------|--------------------------------|
| w/o GM          | 37.1                             | 43.8                            | 38.3                           |
| w/ GM           | <b>42.2</b>                      | <b>44.9</b>                     | <b>40.9</b>                    |
| w/o CA          | 37.7                             | 45.6                            | 40.4                           |
| w/ CA           | <b>42.2</b>                      | <b>44.9</b>                     | <b>40.9</b>                    |
| w/o SA          | 35.4                             | 38.3                            | 36.6                           |
| w/ SA( $N=3$ )  | 39.6                             | 43.9                            | 39.0                           |
| w/ SA( $N=7$ )  | 40.2                             | <b>45.9</b>                     | 40.5                           |
| w/ SA( $N=13$ ) | <b>42.2</b>                      | 44.9                            | <b>40.9</b>                    |
| max pooling     | 37.5                             | 43.1                            | 34.9                           |
| avg pooling     | <b>42.2</b>                      | <b>44.9</b>                     | <b>40.9</b>                    |

**Effectiveness of channel-wise attention.** To verify the effectiveness of channel-wise attention, we conduct two variants to compute the embedded vector  $\mathcal{V}$  in (4), where weighted summation  $\mathcal{V} = \sum_{n=1}^N V^n \cdot w^n$  and equal summation  $\mathcal{V} = \frac{1}{N} \sum_{n=1}^N V^n$  are denoted as “w/ CA” and “w/o CA” respectively. The results are shown in Table 9. Notably, for similar domains (*e.g.*, Sim10k to Cityscapes or Cityscapes to Foggy Cityscapes), we obtain very similar result without channel-wise attention; while for dissimilar domains (*e.g.*, PASCAL to Clipart or PASCAL to Watercolor), we observe an obvious drop in performance, *i.e.*, 4.5% vs. 2.9%. This is maybe because similar/dissimilar domains share similar/different semantic information in each feature map of the spatial pyramid.

## 5 Conclusions

In this work, we propose a general domain adaptation framework for various computer vision tasks including object detection, instance segmentation and semantic segmentation. Given target-specific guided information, our method can make full use of feature maps in the spatial attention pyramid, which enforces the

network to focus on the most discriminative semantic regions for domain adaptation. Extensive experiments conducted on various challenging domain adaptation datasets demonstrate the effectiveness of the proposed, which performs favorably against the state-of-the-art methods.

## References

1. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *TPAMI* **39**(6) (2017) 1137–1149 [1](#), [4](#), [11](#)
2. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. In: *CVPR*. (2018) 4203–4212 [1](#)
3. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: *ICCV*. (2017) 2980–2988 [1](#), [4](#), [9](#), [11](#)
4. Hayder, Z., He, X., Salzmann, M.: Boundary-aware instance segmentation. In: *CVPR*. (2017) 587–595 [1](#)
5. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *CVPR*. (2015) 3431–3440 [1](#)
6. Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. *CoRR* **abs/1706.05587** (2017) [1](#)
7. Quiñero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.: Covariate shift and local learning by distribution matching (2008) [2](#)
8. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR* **abs/1612.02649** (2016) [2](#), [3](#)
9. Liu, M., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: *NeurIPS*. (2017) 700–708 [2](#)
10. Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: *ICML*. (2018) 1994–2003 [2](#)
11. Chen, Y., Li, W., Sakaridis, C., Dai, D., Gool, L.V.: Domain adaptive faster R-CNN for object detection in the wild. In: *CVPR*. (2018) 3339–3348 [2](#), [3](#), [8](#), [9](#), [10](#), [11](#), [12](#)
12. Zhu, X., Pang, J., Yang, C., Shi, J., Lin, D.: Adapting object detectors via selective cross-domain alignment. In: *CVPR*. (2019) 687–696 [2](#), [3](#), [9](#), [10](#), [11](#), [12](#)
13. Tsai, Y., Sohn, K., Schuster, S., Chandraker, M.: Domain adaptation for structured output via discriminative patch representations. *CoRR* **abs/1901.05427** (2019) [2](#), [3](#)
14. Luo, Y., Zheng, L., Guan, T., Yu, J., Yang, Y.: Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In: *CVPR*. (2019) 2507–2516 [2](#), [3](#), [11](#), [13](#)
15. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. In: *CVPR*. (2019) 6956–6965 [2](#), [3](#), [8](#), [9](#), [10](#), [11](#), [12](#)
16. Shen, Z., Maheshwari, H., Yao, W., Savvides, M.: SCL: towards accurate domain adaptive object detection via gradient detach based stacked complementary losses. *CoRR* **abs/1911.02559** (2019) [2](#), [3](#), [9](#), [10](#), [11](#), [12](#)
17. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI* **37**(9) (2015) 1904–1916 [2](#), [4](#)
18. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: *CVPR*. (2016) 3213–3223 [2](#), [9](#), [10](#), [11](#), [12](#)



19. Sakaridis, C., Dai, D., Gool, L.V.: Semantic foggy scene understanding with synthetic data. *IJCV* **126**(9) (2018) 973–992 [2](#), [9](#), [10](#), [11](#)
20. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: *CVPR*. (2012) 3354–3361 [2](#), [9](#), [10](#)
21. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes (VOC) challenge. *IJCV* **88**(2) (2010) 303–338 [2](#), [9](#), [10](#)
22. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In: *CVPR*. (2018) 5001–5009 [2](#), [9](#), [10](#)
23. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In Leibe, B., Matas, J., Sebe, N., Welling, M., eds.: *ECCV*. Volume 9906. (2016) 102–118 [2](#), [11](#), [12](#)
24. Ganin, Y., Lempitsky, V.S.: Unsupervised domain adaptation by backpropagation. In Bach, F.R., Blei, D.M., eds.: *ICML*. Volume 37. (2015) 1180–1189 [3](#), [8](#)
25. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. In: *ICCV*, IEEE Computer Society (2017) 2999–3007 [3](#), [11](#)
26. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: *CVPR*. (2018) 7132–7141 [3](#)
27. Wang, X., Girshick, R.B., Gupta, A., He, K.: Non-local neural networks. In: *CVPR*. (2018) 7794–7803 [3](#)
28. Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks. In: *CVPR*. (2019) 510–519 [3](#), [6](#)
29. Woo, S., Park, J., Lee, J., Kweon, I.S.: CBAM: convolutional block attention module. In Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., eds.: *ECCV*. Volume 11211. (2018) 3–19 [3](#)
30. Wang, X., Li, L., Ye, W., Long, M., Wang, J.: Transferable attention for domain adaptation. In: *AAAI*. (2019) 5345–5352 [3](#)
31. Sun, W., Wu, T.: Learning spatial pyramid attentive pooling in image synthesis and image-to-image translation. *CoRR* **abs/1901.06322** (2019) [3](#)
32. Singh, B., Davis, L.S.: An analysis of scale invariance in object detection SNIP. In: *CVPR*. (2018) 3578–3587 [3](#)
33. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: *CVPR*. (2017) 936–944 [3](#)
34. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI* **40**(4) (2018) 834–848 [4](#), [11](#)
35. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: *CVPR*. (2006) 2169–2178 [4](#)
36. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. (2017) [9](#)
37. Kim, T., Jeong, M., Kim, S., Choi, S., Kim, C.: Diversify and match: A domain adaptive representation learning paradigm for object detection. In: *CVPR*. (2019) 12456–12465 [9](#)
38. He, Z., Zhang, L.: Multi-adversarial faster-rcnn for unrestricted object detection. *CoRR* **abs/1907.10343** (2019) [9](#)
39. Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S.N., Rosaen, K., Vasudevan, R.: Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In: *ICRA*. (2017) 746–753 [9](#), [10](#)
40. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *CVPR*. (2016) 770–778 [10](#)

41. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F.: Imagenet: A large-scale hierarchical image database. In: CVPR. (2009) 248–255 [10](#)
42. Kim, S., Choi, J., Kim, T., Kim, C.: Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. CoRR **abs/1909.00597** (2019) [11](#)
43. Ros, G., Sellart, L., Materzynska, J., Vázquez, D., López, A.M.: The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR. (2016) 3234–3243 [11](#), [12](#)
44. Chen, Y., Li, W., Gool, L.V.: ROAD: reality oriented adaptation for semantic segmentation of urban scenes. In: CVPR. (2018) 7892–7901 [13](#)
45. Tsai, Y., Hung, W., Schuster, S., Sohn, K., Yang, M., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: CVPR. (2018) 7472–7481 [13](#)