

Portland State University

**PDXScholar**

---

Electrical and Computer Engineering Faculty  
Publications and Presentations

Electrical and Computer Engineering

---

11-22-2022

# COCM: Co-Occurrence-Based Consistency Matching in Domain-Adaptive Segmentation

Siyu Zhu

*University of Chinese Academy of Sciences, Beijing*

Yingjie Tian

*University of Chinese Academy of Sciences, Beijing*

Fenfen Zhou

*University of Chinese Academy of Sciences, Beijing*

Kunlong Bai

*University of Chinese Academy of Sciences, Beijing*

Xiaoyu Song

*Portland State University, [song@ece.pdx.edu](mailto:song@ece.pdx.edu)*

Follow this and additional works at: [https://pdxscholar.library.pdx.edu/ece\\_fac](https://pdxscholar.library.pdx.edu/ece_fac)



Part of the [Electrical and Computer Engineering Commons](#)

**Let us know how access to this document benefits you.**

---



## Citation Details

Zhu, S., Tian, Y., Zhou, F., Bai, K., & Song, X. (2022). COCM: Co-Occurrence-Based Consistency Matching in Domain-Adaptive Segmentation. *Mathematics*, 10(23), 4468.

This Article is brought to you for free and open access. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Publications and Presentations by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: [pdxscholar@pdx.edu](mailto:pdxscholar@pdx.edu).

## Article

# COCM: Co-Occurrence-Based Consistency Matching in Domain-Adaptive Segmentation

Siyu Zhu <sup>1</sup> , Yingjie Tian <sup>2,3,4,5,\*</sup> , Fenfen Zhou <sup>1</sup>, Kunlong Bai <sup>1</sup> and Xiaoyu Song <sup>6</sup><sup>1</sup> School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China<sup>2</sup> School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100190, China<sup>3</sup> The Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China<sup>4</sup> The Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China<sup>5</sup> The MOE Social Science Laboratory of Digital Economic Forecasts and Policy Simulation, University of Chinese Academy of Sciences, Beijing 100190, China<sup>6</sup> Department of Electrical and Computer Engineering, Portland State University, Portland, OR 97203, USA

\* Correspondence: tyj@ucas.ac.cn

**Abstract:** This paper focuses on domain adaptation in a semantic segmentation task. Traditional methods regard the source domain and the target domain as a whole, and the image matching is determined by random seeds, leading to a low degree of consistency matching between domains and interfering with the reduction in the domain gap. Therefore, we designed a two-step, three-level cascaded domain consistency matching strategy—co-occurrence-based consistency matching (COCM)—in which the two steps are: Step 1, in which we design a matching strategy from the perspective of category existence and filter the sub-image set with the highest degree of matching from the image of the whole source domain, and Step 2, in which, from the perspective of spatial existence, we propose a method of measuring the PIOU score to quantitatively evaluate the spatial matching of co-occurring categories in the sub-image set and select the best-matching source image. The three levels mean that in order to improve the importance of low-frequency categories in the matching process, we divide the categories into three levels according to the frequency of co-occurrences between domains; these three levels are the head, middle, and tail levels, and priority is given to matching tail categories. The proposed COCM maximizes the category-level consistency between the domains and has been proven to be effective in reducing the domain gap while being lightweight. The experimental results on general datasets can be compared with those of state-of-the-art (SOTA) methods.

**Keywords:** computer vision; semantic segmentation; domain adaptation; image matching**MSC:** 68T05**Citation:** Zhu, S.; Tian, Y.; Zhou, F.; Bai, K.; Song, X. COCM:Co-Occurrence-Based Consistency Matching in Domain-Adaptive Segmentation. *Mathematics* **2022**, *10*, 4468. <https://doi.org/10.3390/math10234468>

Academic Editor: Bo-Hao Chen

Received: 22 October 2022

Accepted: 18 November 2022

Published: 26 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



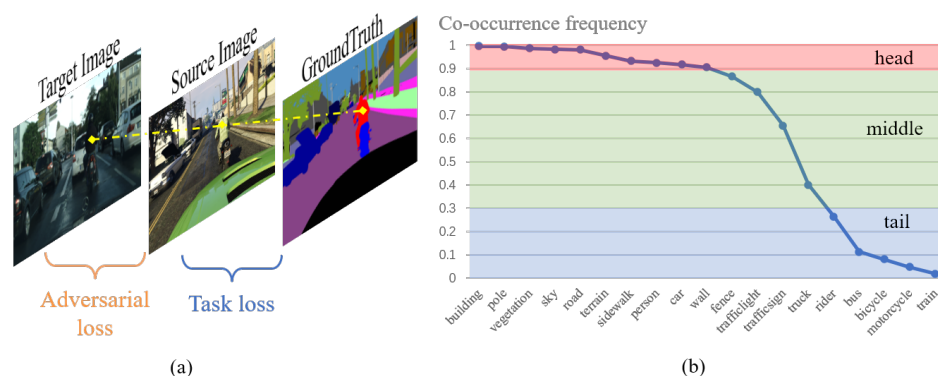
**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Domain-adaptive semantic segmentation (DASS) has received extensive attention. It can greatly alleviate the high cost of manual annotation in intensive prediction. Researchers have made significant progress in exploring methods for adaptation from a labeled source domain to an unlabeled target domain.

Our work pays attention to the DASS method, which is based on adversarial training. Previous methods used vanilla generative adversarial networks (GANs) [1], patch GANs [2], and pixel-level GANs [3]. Here, we focus on the pixel-level GAN method. As shown in Figure 1a, the guidance for each pixel comes from two parts: the adversarial loss and task loss. The former pursues domain-invariant space, while the latter maintains the segmentation performance. The intuitive idea is that if these three pixels in the same

position belong to the same category, they will produce more sufficient guidance. This kind of matching is called semantic-level consistency matching. The traditional DASS method regards the source domain and the target domain as a whole set, and the image matching is determined by randomly specified seeds. This matching method does not consider the semantic-level consistency between domains, which leads to a negative transfer in domain adaptation.



**Figure 1.** (a) A group of images during training. From left to right, they are the target image, source image, and pixel-level ground truth. (b) The imbalanced distribution of the co-occurrence frequency in different categories.

Therefore, we propose a co-occurrence-based consistency matching (COCM) method for the maximization of the inter-domain semantic consistency. COCM is a two-step strategy for matching from coarse to fine, and it selects the optimal source domain corresponding to an image for the target-domain image by using two steps that assess the “more common categories” (existence) and if elements are “in the same position” (space). At the same time, we fully consider the imbalanced distribution of the co-occurrence frequencies of different categories, as shown in Figure 1b. The categories are divided into three levels according to the frequency of the inter-domain co-occurrence of categories—head, middle, and tail, corresponding to the red, green, and blue areas in the figure. The order of priority is in the reverse order of frequency to ensure the contribution of low-frequency categories to the consistency matching.

Previous work related to content-consistent matching (CCM) [4] matched the source-domain image by clustering target images. However, this work was focused more on global matching and lacked consideration of inter-domain co-occurrence. Our method pays attention to differential co-occurrence categories in the consistency matching and adjusts for imbalanced distributions. As shown in Figure 2, the three columns are the target domain image, the image matched by the COCM, and the image matched with the vanilla method, respectively. The three lines show the matching at different levels. We can see that the images in the second column basically met the matching target of the same location that corresponded to the same category, while the semantic consistency matching in the third column was not satisfactory.

Our contributions are:

- We propose a new co-occurrence based consistent matching (COCM) method. To the best of our knowledge, this is the first effort to explore image matching from the perspective of inter domain category co-occurrence frequency.
- The COCM is composed of two-step cascade matching and three-level priority strategy. Two-step refers to matching the optimal source image for the target domain image from the existence and spatial matching. Three-level refers to the priority adjustment of category co-occurrence imbalance.
- We design a new measurement patch intersection over union (PIOU) to measure the spatial similarity between domains.

- Our method is lightweight and proved to be effective. The results on general datasets can compare with SOTA methods.



**Figure 2.** Examples of co-occurrence-based consistency matching on three levels.

## 2. Related Work

### 2.1. Domain Adaptive Semantic Segmentation

Domain adaptive semantic segmentation (DASS) is one of the important applications of domain adaptation. The main purpose of this task is to obtain the optimal segmentation performance for the unsupervised target domain. Some of previous methods used adversarial training to maximize domain invariance from the feature space [1,3] or the label space [5], some introduced style transfer to explore the adaptation of segmentation on the basis of style consistency, and some used self-supervised learning to achieve domain adaptation by exploring more accurate pseudo labels. Ref. [6] proposes the contextual-relation consistent domain adaptation (CrCDA), which explicitly learns and enforces prototype local contextual-relations in the feature space of the labeled source domain and transfers them to the unlabeled target domain by adversarial learning. CrCDA applies co-occurrence frequency from the perspective of local contextual-relation, and our method applies co-occurrence frequency from the perspective of image matching. Ref. [7] proposes the pixel level cycle association (PLCA), which establishes pixel-level cycle association between source and target pixel pairs, and, in contrast, strengthens the connection between them to reduce the domain gap. We are inspired by cycle association and applied to class-level consistency matching between domains.

### 2.2. Image Matching Cross Domain

Cross domain image matching refers to selecting the image closest to the target domain from the source domain according to different standards. In feature distribution matching (FDM) [8], feature distribution matching was proposed to match source domain images from the perspective of color features. The work of [9] performs cross domain image matching from the perspective of outlier detection. CCM [4] selects the positive images in the source domain images. The selection strategy is to cluster the target domain and randomly select 20% of the source images to score with the cluster center of the target domain. Inspired by the above image matching methods, our method matches the entire source domain dataset by two steps, and proposes a new spatial similarity evaluation method, PIOU.

### 2.3. Class-Imbalance Learning

Category-imbalance refers to the situation that the number of training samples in different categories varies greatly [10]. The current research has proposed several solutions, such as re-sampling, cost-sensitive learning, or transfer learning. In DASS task, some studies also pay attention to the imbalance distribution. Class-balanced self-training (CBST) [11] adjusts the category imbalance in the process of generating pseudo labels. Our method comprehensively considers the imbalance of the co-occurrence category inter domain and conduct category rebalancing through three levels of priority.

### 3. Methods

#### 3.1. Preliminary

First, we define the symbols involved in COCM: source domain image  $I_s$  and ground truth  $Y_s$ , target domain image  $I_t$ , feature extractor  $F$ , segmentation module atrous spatial pyramid pooling ( $ASPP$ ), domain discriminator  $D$ , wherein the segmentation network is composed of  $F + ASPP$ .  $H$ ,  $W$ , and  $C$  denote height, width, and category number, respectively. Vanilla DASS algorithm based on adversarial training, as shown in Figure 3, can be regarded as three steps in training:

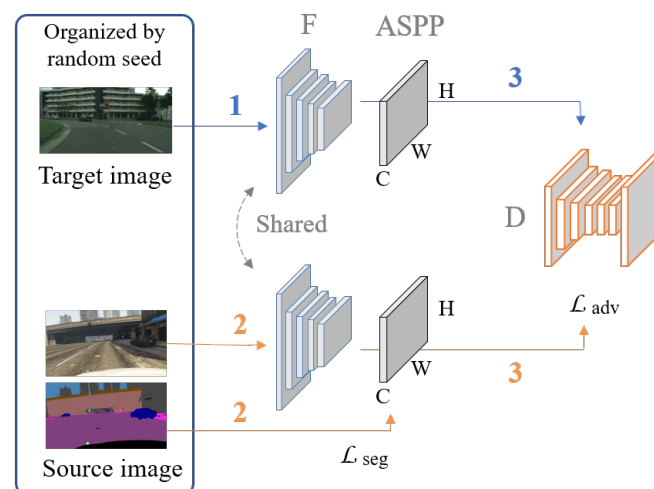
1. The target domain image  $I_t$  is input into  $F$  to obtain segmentation header  $F(I_t)$  and input  $F(I_t)$  into  $ASPP$  to output prediction  $P(I_t)$ .
2. The source domain image  $I_s$  is input into the segmentation network, and the segmentation header  $F(I_s)$  and the result  $P(I_s)$  are output. For  $P(I_s)$ , the cross-entropy loss is used to maintain the performance of the segmentation network. The calculation of the cross-entropy loss is as follows:

$$\mathcal{L}_{seg} = - \sum_{h=1}^H \sum_{w=1}^W \sum_{c=1}^C y_s^{h,w,c} \log P_{x_s}^{h,w,c}. \quad (1)$$

3. The segmentation header  $F(I_t)/F(I_s)$  of the source domain and the target domain is input to the domain discriminator  $D$ . The function of discriminator is to narrow the distribution of source domain and target domain, and maximize the shared information between domains. The discriminator uses adversarial loss as follows:

$$\mathcal{L}_{adv} = -\mathbb{E}[\log D(F(I_s))] - \mathbb{E}[\log(1 - D(F(I_t)))]. \quad (2)$$

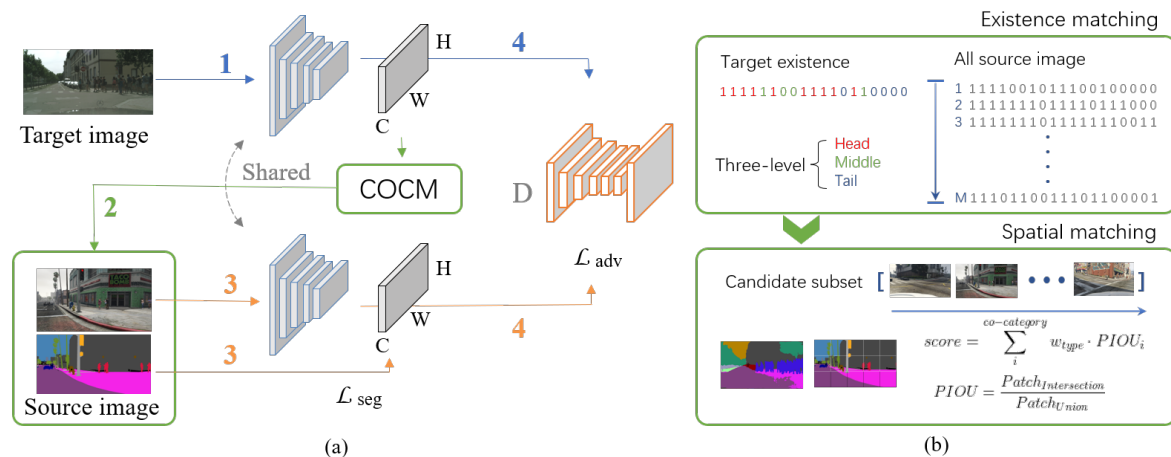
Vanilla methods regard the source domain and the target domain as a whole set, and the image matching is determined by specified random seed with dataloader module. Here ‘regard the source domain and the target domain as a whole’ means that when dealing with domain gap, the entire source domain image set is considered as a uniform distribution of categories, and each source domain image plays an equally important role in adaptation. In model training, source domain and target domain images are randomly selected as input of model. In COCM, the target domain still uses dataloader to set the order, while the source domain image is matched according to the target image prediction.



**Figure 3.** Vanilla DASS training flow based on adversarial training. Step 1, 2, and 3 in the figure are explained in detail in Section 3.1.

### 3.2. Co-Occurrence Based Consistent Matching

We show the overall frame of COCM in Figure 4. We can see from (a) that compared with the vanilla method in Figure 3, we added the COCM module to match the source domain image. The process of domain adaptation training after matching is the same as that of vanilla method. In (b), we show the existence matching and spatial matching, and next we will explain it in detail.



**Figure 4.** Overall frame of co-occurrence based consistent matching (COCM). The left side is the proposed training flow. Our proposed COCM follows step 1 in the vanilla DASS algorithm as step 2. Therefore, steps 2 and 3 in vanilla DASS algorithm become steps 3 and 4. The right side shows the details of COCM, and from top to bottom are the two steps of existence and spatial matching.

#### 3.2.1. Existence Matching

Due to the domain gap, it cannot be guaranteed that each pair of matched images contains the same category. Therefore, we first performed existence matching to screen out the subset with the highest degree of co-occurrence.

For the target domain image, we obtain a one-hot vector  $E_{tgt}$  indicating whether the category exists according to the current prediction result  $P(I_t)$  as shown in Figure 4b. The vector is in  $1 \times C$  dimension. If the corresponding bit is 1, the category exists in the image. If the bit is 0, the category does not exist. For the source domain, we obtain the category existence information of all images according to the Ground Truth, and each image corresponds to a heat vector. The entire dataset is pre-generated in the form of matrix  $M_{src}$ .

For category level existence matching, we traverse the target domain existence vector  $E_{tgt}$  across the source domain matrix  $M_{src}$  to find the candidate subset with the highest matching degree. We design matching strategy with three-level priority. According to the co-occurrence frequency distribution inter domain mentioned in Figure 1, the categories are divided by threshold into three levels: head, middle and tail. In matching, if there is at least one common category in the tail level, it is marked as tail level matching and the number of common categories is counted. If the tail level does not match at all, then retrieve and find whether the middle level has at least one common category and count them. If the middle level does not match at all as well, retrieve and find whether the head level has at least one common category and count them.

The existential matching outputs the candidate image set with the largest number of common categories in marked matching level. Existence matching performs coarse preliminary screening from the entire target domain data set, and the following spatial matching performs fine selection from the perspective of location.

#### 3.2.2. Spatial Matching

Spatial matching aims to achieve the goal of the same location and the same category to the maximum extent. Different from the traditional pixel-level image matching, cross



domain image matching looks for the image with the same overall layout and the same position of low-frequency category. Therefore, ‘the same position’ in COCM refers to one patch. Inspired by the classical measurement method MIOU, we propose the method of PIOU to quantitatively measure the spatial similarity of common categories. The scoring method is as follows:

$$score = \sum_i^{co-category} w_{type} \cdot PIOU_i, \quad (3)$$

where  $i$  refers to the category of co-occurrence,  $PIOU_i$  means Patch Intersection-over-Union of each category, and  $w_{type}$  is used as a hyper parameter to adjust the importance of low-frequency category in matching.

We partitioned the image into patches. For the target image, we divide it into  $H/N \times W/N$  patches according to the current prediction result  $P(I_t)$ , number each position from  $0 \sim H/N \times W/N - 1$ , and count the patch number covering each category. Where  $N$  is a hyper parameter to adjust the size of the patch. For the source images, we divide the patches in advance according to the ground truth and record the category space information of all images.

For all co-occurrence categories, we calculate and sum the intersection and union ratio of the covering patch blocks on the target domain and the source domain. To highlight the contribution of low-frequency categories to the total score, we adjust it by different weights. Finally, we output the image with the highest score as the corresponding source domain image of the current target domain image. To avoid repeatedly selecting the same source domain image. We recorded the selected images of each target domain image and excluded them from the candidate list before each match. The calculation of PIOU is as follows:

$$PIOU = \frac{Patch_{Intersection}}{Patch_{Union}}. \quad (4)$$

To better present the calculation of PIOU, we chose a set of representative images in Figure 5. By existence matching, we know the co-occurrence categories are rider and motorcycle. Here we calculate the scores of rider. The upper left is the target domain image, the lower left is the image to be scored in the source domain, the upper middle is the prediction result of the target domain after patch division with  $N = 6$ , and the lower middle is the ground truth of the source domain image after patch division. We use black lines to mark the patch division in the middle images. The area marked by white lines is the area covered by the rider in the prediction results and the ground truth. The upper right blue line represents the intersection area, and the lower right yellow line represents the union area. Here we use the number of blocks to measure the spatial matching of rider, so  $PIOU_{rider} = 6/8(0.75)$ .

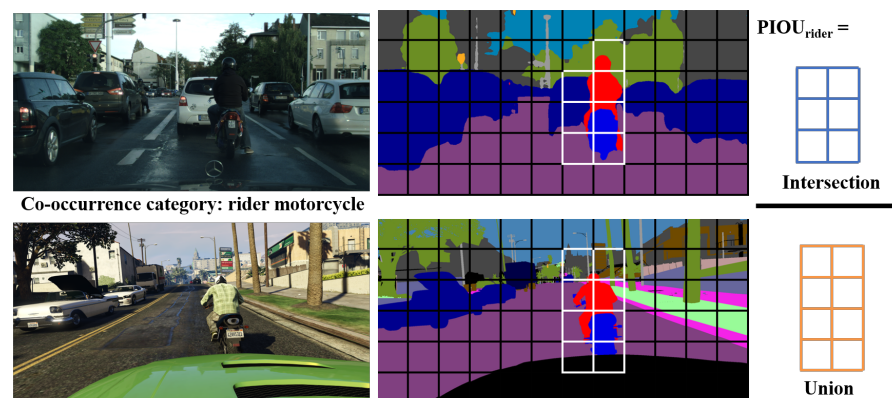


Figure 5. Visualization of PIOU calculation.

We choose the patch segmented image for spatial matching for the following reason: Firstly, the computational cost of pixel level position matching is too high, which seriously affects the selection efficiency. Second, the prediction result of the target domain is not completely accurate. Patch level matching can produce a certain fault tolerance for wrong pixel matching. Third. Patch matching is a relaxation strategy. As long as the prediction results indicate the approximate positions of the co-occurrence categories, the COCM can match their corresponding images in the source domain.

### 3.2.3. Training Procedure of COCM

The training procedure of our proposed method is summarized in Algorithm 1.

---

#### Algorithm 1: Training procedure of COCM.

---

**Input:** The source image set  $I_s$  and ground truth  $Y_s$ ; The target image set  $I_t$ ; The source-domain parameter  $\theta_s$ ; The iteration number  $T$ ; Threshold of head/middle/tail.

**Output:** Adapted target-domain segmentation network parameter  $\theta_T$

- 1: Train source domain supervised and share  $\theta_s$  with  $\theta_T$ .
  - 2: Use  $Y_s$  to calculate the existence information  $M_{exist}$  and spatial information  $M_{space}$  of source image set.
  - 3: Initialize category co-occurrence frequency with source domain category frequency.
  - 4: Generate the target domain image iterator  $T_{Iter}$  with random seed.
  - 5: **for** iteration 1 to  $T$  **do**
  - 6:    $T_{Iter}$  obtains the current batch of  $I_t$  and input the segmentation network  $F + ASPP$  to obtain  $P(I_t)$ .
  - 7:   **Existence matching:** calculate category existence vector  $V(I_t)$  of  $P(I_t)$ , and traverse  $M_{Exist}$  to find the subset  $I_{SUB}$  with the most matching digits according to the priority of tail > middle > head.
  - 8:   **Spatial matching:** calculate category location tuple  $T(I_t)$  and traverse  $I_{SUB}$ . Find image set  $I_{MAX}$  with the highest score according to Formula (3) and  $M_{space}$  and randomly select one of them as the corresponding image  $I_s$ .
  - 9:    $I_s$  input the segmentation network  $F + ASPP$  to obtain  $P(I_s)$ , and calculate segmentation loss in Formula (1).
  - 10:    $I_t$  and  $I_s$  input the discriminator  $D$ , and calculate the adversarial loss in Formula (2).
  - 11:   By alternately training  $F + ASPP$  and  $D$ ,  $F + ASPP$  is encouraged to generate domain-invariant features.
  - 12:   Update the frequency of inter domain category co-occurrence after a fixed iteration.
  - 13: **end for**
  - 14: **return**  $\theta_T$
- 

## 4. Experiments

### 4.1. Datasets

Following the general data set of DASS, we choose the Cityscapes [12] as target domain. The data set contains 2975 training images and 500 validation images. Images were collected from more than 50 cities including Aachen, Bochum, and Bremen, reaching  $1024 \times 2048$  resolution. The image set has 30 predefined categories, 19 of which are used in the semantic segmentation task.

For source set we use GTA5 [13] and SYNTHIA [14] datasets. GTA5 dataset obtains street view images from the classic commercial game GTAV, and generates a large number of high-resolution annotation images by computer graphics technology. GTA5 contains 24,966 images with a resolution of  $1914 \times 1052$ . The image set predefines 19 categories to match Cityscapes. SYNTHIA is an urban street view data set generated by the Utility development tool, with a resolution of  $1280 \times 960$ . Here we use the subset SYNTHIA-RAND-CITYSCAPES because its annotation space corresponds to cityscapes. The total number of images reached 9400.



#### 4.2. Implementation Details

The backbone of the segmentation network adopts the ResNet-101 [15] model based on DeepLab-V2 [16] structure. We use the segmentation model pre-trained on ImageNet [17] in the initial state. For the segmentation network, our structure includes five convolution layers, with a convolution core of 4, the number of channels of  $\{64, 128, 256, 512, 1\}$ , a step size of 2, which is similar to the structure of AdaptSegNet [5]. Following the training setting of AdaptSegNet [5], the optimizer of the feature extractor uses SGD [18], the momentum value is 0.9, the weight decay value is  $10 \times 10^{-4}$ , the initial learning rate is  $2.5 \times 10^{-4}$ , and the poly learning rate policy is used for attenuation. For the discriminator, our structure consists of three convolution layers, with a convolution kernel of 3, the number of channels of  $\{256, 128, 2C\}$  ( $C$  refers to the number of categories) which is similar to the structure of [3], and the step size of 1. The optimizer of discriminator uses Adam [19], where  $\beta_1 = 0.9$ ,  $\beta_2 = 0.99$ , the weight decay value is set to  $10 \times 10^{-4}$ , the initial learning rate is  $2.5 \times 10^{-4}$ , and the poly learning rate policy is used for attenuation.

We set the batch size to 6 in  $\text{GTA5} \rightarrow \text{Cityscapes}$  and 4 in  $\text{SYNTHIA} \rightarrow \text{Cityscapes}$ , respectively. The crop size in the target domain is set to  $1024 \times 512$ , in the source domain are set to  $1280 \times 760$ . Hyperparameter  $\lambda_{adv}$  set to 0.01. The thresholds of  $\{head, middle, tail\}$  were set to  $\{0.9, 0.3\}$  in  $\text{GTA5} \rightarrow \text{Cityscapes}$  and  $\{0.9, 0.5\}$  in  $\text{SYNTHIA} \rightarrow \text{Cityscapes}$ , respectively. We have a supervised training source domain in advance and serve as the initialization of domain adaptation and update the frequency of inter domain category co-occurrence every 2000 iteration. To further improve the performance, we used self-distillation [20] with multi-scale in testing stage. Our experiment is implemented in the Pytorch library on a GTX 3090 with 24 GB memory.

For the evaluation metrics, we use the commonly used evaluation metrics in DASS: Mean Intersection-over-Union(MIOU) [21]. Where Intersection-over-Union (IOU) evaluates the accuracy of the corresponding class, and MIOU calculates the average value of IOU.

#### 4.3. Quantitative Comparison Studies

We compare the excellent work based on adversarial training: AdaptSegNet [5], CLAN [1], CAG-UDA [22], FADA [3], PAM [23], based on inter domain consistency: CCM [4], CrCDA [6], PLCA [7]. We listed the performance of source only and full supervision (Oracle) for reference.

##### 4.3.1. From GTA5 Adapt to Cityscapes

In Table 1, we can see that the performance of our method can reach 51.1% MIOU on the validation set and 52.6% MIOU on the test set, of which eight categories are optimal. Those categories with high-frequency co-occurrence, such as road, building, vegetation, terrain, sky, person, and car have achieved results close to oracle, which proves the effectiveness of our method. Meanwhile, low-frequency co-occurrence categories, such as rider and motor, have also made significant improvements. We also observed that adaptation in some categories with low-frequency co-occurrence, such as traffic sign or train, was not satisfactory. We found that the feature similarity between these categories is very low, which is difficult to distinguish by human eyes. Even the bus in GTA5 dataset looks more similar to the train in Cityscapes dataset. We consider that image semantic matching cannot improve the performance of feature dissimilar categories and unseen categories cross domain, but can significantly improve the performance of similar features. This is also the disadvantage of consistent image matching between domains.

**Table 1.** Quantitative comparison on GTA5 → cityscapes task.

Method	MIoU	Road	Side.	Buil.	Wall	Fence	Pole	Tlight	Tsign	Vege.	Terr.	Sky	Person	Rider	Car	Truck	Bus	Train	Motor	Bike
Source only	36.2	63.0	14.5	68.7	23.0	17.4	21.6	34.4	11.0	82.5	22.0	76.0	55.0	32.0	58.7	24.6	29.3	16.3	26.9	11.3
AdaptSegNet	42.4	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1
PLCA	47.7	84.0	30.4	82.4	35.3	24.8	32.2	36.8	24.5	85.5	37.2	78.6	66.9	32.8	85.5	40.4	48.0	8.8	29.8	<b>41.8</b>
CrCDA	48.6	92.4	55.3	82.3	31.2	29.1	32.5	33.2	35.6	83.5	34.8	84.2	58.9	32.2	84.7	<b>40.6</b>	46.1	2.1	31.1	32.7
CCM	49.9	91.9	44.7	82.6	29.4	19.4	30.2	37.4	28.9	82.1	44.8	84.9	61.8	31.6	83.3	23.9	42.2	0.9	28.6	29.5
FADA	50.1	91.0	50.6	86.0	<b>43.4</b>	<b>29.8</b>	36.8	<b>43.4</b>	25.0	86.8	38.3	87.4	64.0	38.0	85.2	31.6	46.1	6.5	25.4	37.1
CAG-UDA(val)	50.2	90.4	51.6	83.8	34.2	27.8	<b>38.4</b>	25.3	<b>48.4</b>	85.4	38.2	78.1	58.6	34.6	84.7	21.9	42.7	<b>41.1</b>	29.3	37.2
COCM(val)	51.1	93.8	54.5	86.4	41.6	28.0	32.4	42.1	29.5	86.5	37.3	88.1	63.8	35.9	87.7	38.6	<b>50.2</b>	18.0	28.7	27.8
CAG-UDA(test)	51.7	93.2	<b>57.0</b>	85.6	35.7	25.1	37.5	30.8	45.3	87.1	50.1	89.4	62.7	40.8	87.8	18.0	32.4	34.5	34.4	35.4
PAM(test)	52.0	92.8	47.5	86.0	36.3	15.4	29.9	41.0	21.4	86.8	51.0	87.5	68.1	<b>45.0</b>	88.6	30.3	41.3	<b>41.1</b>	<b>44.7</b>	33.6
COCM(test)	<b>52.6</b>	<b>94.5</b>	54.2	<b>86.8</b>	36.1	21.4	31.9	42.4	28.6	<b>88.1</b>	<b>51.1</b>	<b>91.5</b>	<b>69.6</b>	<b>45.0</b>	<b>89.9</b>	33.1	39.8	27.7	40.3	27.9
Oracle	66.8	96.7	75.0	88.5	51.0	46.7	39.0	47.4	58.6	88.3	53.0	91.6	67.4	46.9	90.7	68.6	76.1	67.9	51.2	63.9

#### 4.3.2. From SYNTHIA Adapt to Cityscapes

Table 2 shows the quantitative comparison on SYNTHIA  $\rightarrow$  cityscapes task. Our proposed method can achieve an accuracy of 52.7% MIoU in the validation set, in which the road category is optimal and the performance of high-frequency categories such as sidewalk, traffic sign, sky, car, and motor ranked second. This is basically in line with our expectations. The overall adaptation performance of SYNTHIA data set is not as good as that of GTA5. We believe that it is due to the relatively low degree of realism of image set.

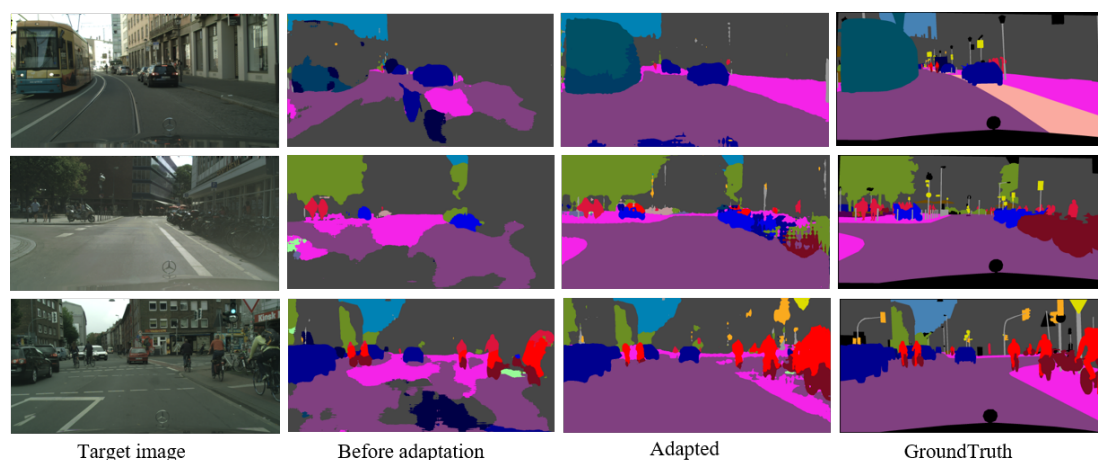
**Table 2.** Quantitative comparison on SYNTHIA  $\rightarrow$  cityscapes task.

Method	MIoU	Road	Side.	Buil.	Tlight	Tsign	Vege.	Sky	Person	Rider	Car	Bus	Motor	Bike
Source only	32.0	52.5	21.5	68.8	10.7	11.4	75.7	73.9	49.3	4.9	73.2	30.0	4.2	7.7
AdaptSegNet	46.7	84.3	42.7	77.5	4.7	7.0	77.9	82.5	54.3	21.0	72.3	32.2	18.9	32.3
CLAN	47.8	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
CrCDA	50.0	86.2	<b>44.9</b>	79.5	9.4	11.8	78.6	<b>86.5</b>	57.2	26.1	76.8	39.9	21.5	32.1
PAM	52.4	84.3	39.3	82.8	6.9	14.9	85.4	85.5	58.4	27.9	84.1	<b>49.3</b>	27.1	35.0
FADA	52.5	84.5	40.1	83.1	20.1	<b>27.2</b>	84.8	84.0	53.5	22.6	<b>85.4</b>	43.7	26.8	27.8
CCM	52.6	79.6	36.4	80.6	<b>22.4</b>	14.9	81.8	77.4	56.8	25.9	80.7	45.3	<b>29.9</b>	<b>52.0</b>
CAG-UDA	52.6	84.8	41.7	<b>85.5</b>	13.7	23.0	<b>86.5</b>	78.1	<b>66.3</b>	<b>28.1</b>	81.8	21.8	22.9	49.0
COCM	<b>52.7</b>	<b>87.6</b>	43.3	82.9	5.9	25.2	85.1	85.7	54.9	21.7	84.1	39.6	28.8	40.1
Oracle	72.5	96.7	75.0	88.5	47.4	58.6	88.3	91.6	67.4	46.9	90.7	76.1	51.2	63.9

#### 4.4. Qualitative Comparison Studies

##### 4.4.1. Overall Segmentation Performance

Figure 6 shows quantitative adaptation results on GTA 5  $\rightarrow$  cityscapes task. Here we selected three representative images to cover different categories. For each column, we show target image, source only result, adapted result with COCM, and ground truth image from left to right, respectively. For each row, we aim to show the domain adaptation visualization of categories of bus, bicycle, sidewalk, and traffic light. The first row shows the optimization effect of bus category. Our method can cover a large area of bus, although there is still some confusion with train category. It can be seen from the second row that the bicycle is separated after adaptation, although it still overlaps the car to a certain extent. The performance of traffic lights in the third row has been significantly improved, and even the outline is clear. In addition, we can see from the three rows that the segmentation of road and sidewalk is smoother and more coherent. The above is consistent with our previous analysis.



**Figure 6.** (Best viewed in color.) Qualitative adaptation results from GTA5  $\rightarrow$  cityscapes task.

#### 4.4.2. Image Matching Visualization

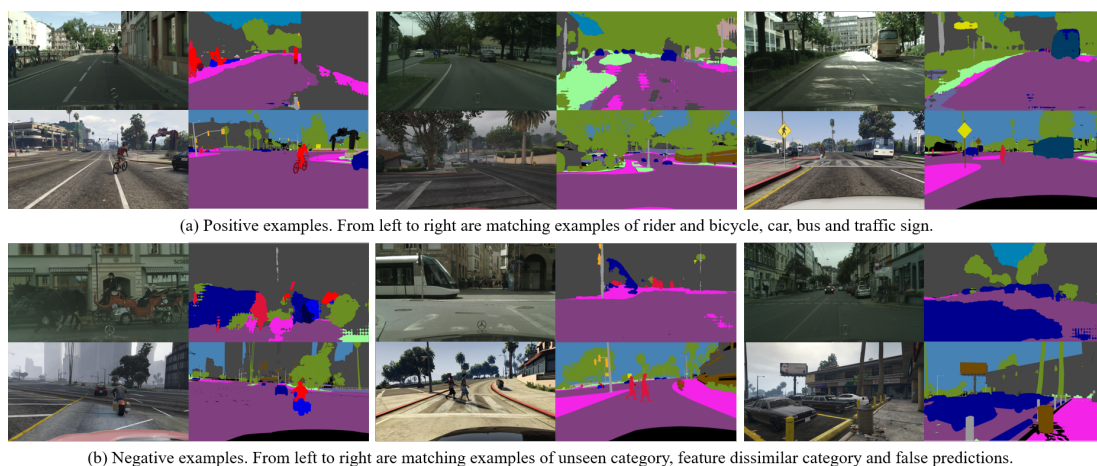
In Figure 7, we show an image of the target domain and the corresponding image of the source domain selected for five consecutive times. In the first row, the target domain image contains the tail level category rider, which appears in the corresponding positions of the following five graphs. The second row shows the image matching of car category, which basically realizes the goal of same category in same location, but there is a confused matching in the third image matching the truck in the corresponding location. The third row shows the matching of the bus category. We can see that the bus appears in the corresponding position of the first, second, fourth, and fifth source images, and the bus appears on the left of the image in the third. The overall layout structure of the above images, including road, building, and sky, is roughly similar. The tail category basically appears in the corresponding position, which is in line with our expectation.



**Figure 7.** Target domain image and the five corresponding images matching by COCM.

#### 4.4.3. Matching Examples

To further illustrate the advantages and disadvantages of the COCM method, we selected a group of positive matching examples and negative matching examples for visualization. Positive examples can be seen in Figure 8a, the prediction results (upper right of each group of figures) can indicate their approximate positions, and the COCM can match the images of their source domain counterparts. This is in line with our expectations.



**Figure 8.** Positive matching examples and negative matching examples.

Figure 8b shows negative examples. From left to right, is the unseen category, the feature dissimilar category, and false prediction. We can see that the coach in the left part is an unseen category for the source domain, and the images in the source domain are not able to provide guidance. In the middle part, train represent categories with dissimilar features with source domain. The train in the GTA5 domain is dissimilar with the train in cityscapes, but is similar to the bus in cityscapes. Therefore, the prediction result is not insufficient to guide the COCM to match the appropriate source domain image. The right part shows the situation of prediction error on car. Since large-area roads are incorrectly predicted to be car,

COCM is guided to match the source domain image containing large-area of car. Among these three negative examples, only the false prediction can be corrected with the increase in training epoch. This is consistent with the performance in quantitative experiments. For unseen category, feature dissimilar category, we need to further adapt by means of multi-source domain and few-shot learning. This is the weakness of COCM and our future research direction.

#### 4.5. Ablation and Parameter Studies

We performed ablation experiments on GTA5  $\rightarrow$  cityscapes adaptative semantic segmentation task to verify each components, respectively, and ablation results are shown in Table 3. Here, PIOUS is divided into EM and SM, SP denotes source domain pre-training, EM means existence matching SM represents spatial matching, and SD represents self-distillation strategy. We can see that only EM has achieved an improvement of 10.1% MIOU and only SM has achieved an improvement of 8.3% MIOU. EM + LM can achieve an improvement of 11.8% MIOU, and further combined with SD can achieve 14.9% MIOU, which verifies the effectiveness of our method.

**Table 3.** Ablation study on GTA5  $\rightarrow$  cityscapes task.

SP	EM	SM	SD	MIOU (%)
✓				36.2
✓	✓			46.3
✓		✓		44.5
✓	✓	✓		48.0
✓	✓	✓	✓	51.1

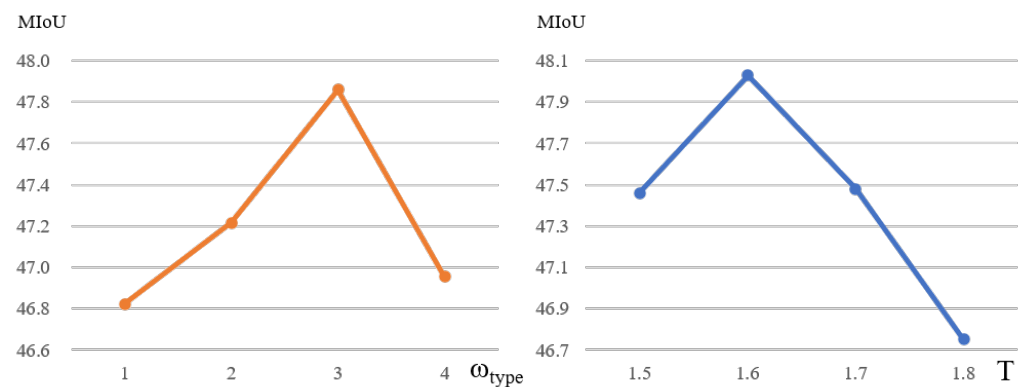
To study the impact of patch partition size on COCM performance, we tested different patch sizes. It can be seen from Table 4, the smaller  $N$  is, the larger the corresponding patch size, and the smaller the number of patches of a single image. We observe that the larger  $N$  is, the smaller the proportion of selected images in the source domain is. When  $N = 6$ , the performance reaches the optimal.

**Table 4.** Parameter study of patch partition on GTA5  $\rightarrow$  cityscapes adaptative semantic segmentation task.

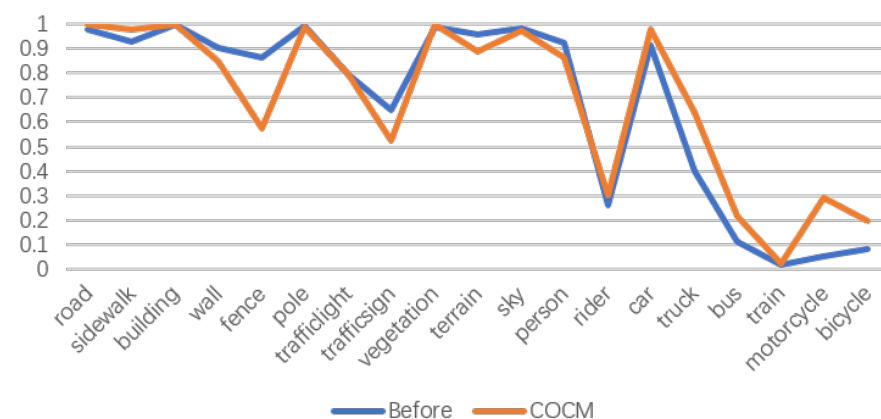
N	Path Size	Count	Source Portion (%)	MIOU (%)
4	$128 \times 128$	32	51.2	46.6
5	$103 \times 103$	50	50.9	46.9
6	$86 \times 86$	72	50.7	48.0
7	$74 \times 74$	98	49.8	47.4

Further, we discuss the weight parameter  $w_{type}$ , which represents the contribution of non-head categories in PIOUS score. As shown in Figure 9, with the increase in non-head category contribution, the adaptive performance is improved and reach optimal when  $w_{type}$  is 3. Following the work of FADA, we studied the effect of parameter  $T$ , which represents the degree of smoothness of the distribution of the prediction results over the categories. As can be seen from right Figure 9, it reaches the optimum when  $T$  is 1.6.

Moreover, we counted the changes in the co-occurrence frequency of category existence before and after the use of COCM on the GTA5  $\rightarrow$  cityscapes task. In Figure 10, we can see that our method has significantly improved in almost the tail category. This is consistent with our previous analysis and proves the effectiveness of our method.



**Figure 9.** Parameter study on  $w_{type}$  and  $T$ . Here  $w_{type} = 1$  means equally important, and 2, 3, and 4 indicate different degrees of importance.



**Figure 10.** Category co-occurrence comparison. The blue line represents the vanilla method and the yellow line represents the COCM method.

## 5. Conclusions

Our method focuses on the category level consistent matching of inter domain images, and designs a three-level two-step cascade matching strategy COCM to select images that meet the same location and category to the maximum extent. In this process, we deal with co-occurrence categories in an imbalance way, and propose a measurement method of  $PIOU$  in spatial matching. Our method effectively improves the class level co-occurrence between domains. Experiments prove that we reduce the domain gap on most semantic categories. At the same time, we also analyze the disadvantage of our method. The effect on unseen categories and feature dissimilar categories is not satisfactory. Therefore, in the future work, we can further improve by means of multi-source domain and few-shot learning. The multi-source domain method can compensate the unseen categories in a single source domain and improve the guidance from the source. The few-shot learning can correct the feature dissimilar categories in the source domain and adjust the deviation generated by the model.

**Author Contributions:** Methodology, S.Z.; validation, F.Z.; writing—original draft preparation, S.Z.; writing—review and editing, Y.T. and X.S.; visualization, K.B.; supervision, Y.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data presented in this study are openly available in [12–14].



**Acknowledgments:** This work has been partially supported by grants from: National Natural Science Foundation of China (No.12071458, 71731009).

**Conflicts of Interest:** The authors declare that they have no conflict of interest to report regarding the present study.

## References

1. 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 Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 2502–2511.
2. Tsai, Y.H.; Sohn, K.; Schuster, S.; Chandraker, M. Domain Adaptation for Structured Output via Discriminative Patch Representations. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, South Korea, 27 October–2 November 2019; pp. 1456–1465. [\[CrossRef\]](#)
3. Wang, H.; Shen, T.; Zhang, W.; Duan, L.; Mei, T. Classes Matter: A Fine-Grained Adversarial Approach to Cross-Domain Semantic Segmentation. In Proceedings of the Computer Vision—ECCV 2020—16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 642–659.
4. Li, G.; Kang, G.; Liu, W.; Wei, Y.; Yang, Y. Content-Consistent Matching for Domain Adaptive Semantic Segmentation. In Proceedings of the Computer Vision—ECCV 2020—16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 440–456.
5. Tsai, Y.H.; Hung, W.C.; Schuster, S.; Sohn, K.; Yang, M.H.; Chandraker, M. Learning to Adapt Structured Output Space for Semantic Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7472–7481.
6. Huang, J.; Lu, S.; Guan, D.; Zhang, X. Contextual-Relation Consistent Domain Adaptation for Semantic Segmentation. In Proceedings of the Computer Vision—ECCV 2020—16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 705–722.
7. Kang, G.; Wei, Y.; Yang, Y.; Zhuang, Y.; Hauptmann, A.G. Pixel-Level Cycle Association: A New Perspective for Domain Adaptive Semantic Segmentation. In Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, Virtual, 6–12 December 2020.
8. Abramov, A.; Bayer, C.; Heller, C. Keep it Simple: Image Statistics Matching for Domain Adaptation. *arXiv* **2020**, arXiv:2005.12551.
9. Liu, X.; Khademi, S.; van Gemert, J.C. Cross Domain Image Matching in Presence of Outliers. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 3250–3256.
10. Zhang, Y.; Kang, B.; Hooi, B.; Yan, S.; Feng, J. Deep Long-Tailed Learning: A Survey. *arXiv* **2021**, arXiv:2110.04596.
11. Zou, Y.; Yu, Z.; Kumar, B.V.K.V.; Wang, J. Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-training. In Proceedings of the Computer Vision—ECCV 2018—15th European Conference, Munich, Germany, 8–14 September 2018; pp. 297–313.
12. 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 Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
13. Richter, S.R.; Vineet, V.; Roth, S.; Koltun, V. Playing for Data: Ground Truth from Computer Games. In Proceedings of the Computer Vision—ECCV 2016—14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Lecture Notes in Computer Science; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9906, pp. 102–118. [\[CrossRef\]](#)
14. 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 Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 3234–3243. [\[CrossRef\]](#)
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
16. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *40*, 834–848. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; Li, F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, FL, USA, 20–25 June 2009; pp. 248–255. [\[CrossRef\]](#)
18. Bottou, L. Large-Scale Machine Learning with Stochastic Gradient Descent. In Proceedings of the 19th International Conference on Computational Statistics, COMPSTAT, Paris, France, 22–27 August 2010; pp. 177–186. [\[CrossRef\]](#)
19. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
20. Yim, J.; Joo, D.; Bae, J.; Kim, J. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 7130–7138. [\[CrossRef\]](#)

21. Garcia-Garcia, A.; Orts, S.; Oprea, S.; Villena-Martinez, V.; Rodriguez, J.A. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv* **2017**, arXiv:1704.06857.
22. Zhang, Q.; Zhang, J.; Liu, W.; Tao, D. Category Anchor-Guided Unsupervised Domain Adaptation for Semantic Segmentation. In Proceedings of the NeurIPS 2019, 8–14 December 2019, Vancouver, BC, Canada, 2019, pp. 433–443.
23. Tian, Y.; Zhu, S. Partial Domain Adaptation on Semantic Segmentation. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 3798–3809. [[CrossRef](#)]