Crowd counting via unsupervised cross-domain feature adaptation

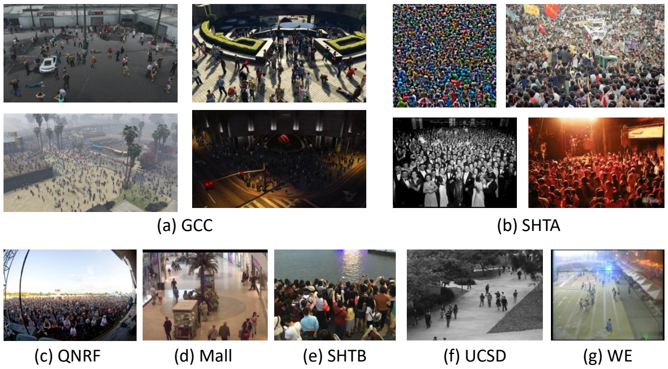
Guanchen Ding, *Student Member, IEEE,* Daiqin Yang, *Member, IEEE,* Tao Wang, Sihan Wang,
and Yunfei Zhang

***Abstract*—Given an image, crowd counting aims to estimate the amount of target objects in the image. With un-predictable installation situations of surveillance systems (or other equipments), crowd counting images from different data sets may exhibit severe discrepancies in viewing angle, scale, lighting condition, etc. As it is usually expensive and time-consuming to annotate each data set for model training, it has been an essential issue in crowd counting to transfer a well-trained model on a labeled data set (source domain) to a new data set (target domain). To tackle this problem, we propose a cross-domain learning network to learn the domain gaps in an unsupervised learning manner. The proposed network comprises of a Multi-granularity Feature-aware Discriminator (MFD) module, a Domain-invariant Feature Adaptation (DFA) module, and a Cross-domain Vanishing Bridge (CVB) module to remove domain-specific information from the extracted features and promote the mapping performances of the network. Unlike most existing methods that use only Global Feature Discriminator (GFD) to align features at image level, an additional Local Feature Discriminator (LFD) is inserted and together with GFD form the MFD module. As a complement to MFD, LFD refines features at pixel level and has the ability to align local features. The DFA module explicitly measures the distances between the source domain features and the target domain features and aligns the marginal distribution of their features with Maximum Mean Discrepancy (MMD). Finally, the CVB module provides an incremental capability of removing the impact of interfering part of the extracted features. Several well-known networks are adopted as the backbone of our algorithm to prove the effectiveness of the proposed adaptation structure. Comprehensive experiments demonstrate that our model achieves competitive performance to the state-of-the-art methods. Code and pre-trained models are available at https://github.com/gcding/CDFA-pytorch.**

***Index Terms*—Unsupervised Crowd Counting, Domain Adaptation, Density Map Estimation, Adversarial Learning**

I. Introduction

COUNTING the number of objects in a given scene has attracted a lot of attentions in recent years. Crowd counting is an important research field in counting, and it has a wide range of applications in video surveillance, traffic control, traffic analysis, etc. During the last half-decade, Convolutional Neural Network (CNN) based methods have made significant progresses and become the mainstream in the field of crowd counting [1]–[7].


(a) GCC

(b) SHTA

(c) QNRF

(d) Mall

(e) SHTB

(f) UCSD

(g) WE

Fig. 1: Sample images from different data sets. The two data sets shown in the first row are used as the source domain, and the five data sets in the second row are used as the target domain in the experiments.

However, a CNN-based crowd counting algorithm relies on a well-labeled data set that contains various scenarios. There are many different types and sizes of public data sets for model training and performance evaluation, such as Shanghai Tech A / B [8], WorldExpo'10 [9], UCF-QNRF [10], etc. Some sample images of these public data sets are shown in Fig. 1. Unfortunately, it is very difficult and time-consuming to construct a data set covering all kinds of scenarios due to innumerable variations in viewing angle, environment condition, equipment capability, etc. When a model, which is well-trained with a labeled data set (i.e., source domain) is taken into the wild (i.e., target domain), it either endures obvious performance degradation or requires time and money consuming new labeled data for fine-tuning of the model. This phenomenon seriously hinders the application of crowd counting algorithms in various practical scenarios.

To deal with this issue, some semi-supervised methods and unsupervised methods are proposed for crowd counting. The key point of semi-supervised learning methods [11], [12] is to transfer a pre-trained model from the source domain to a target domain through a small amount of labeled target domain data. Reddy et al. [11] propose a model in which parameters are learned in a way that facilitates effective fine-tuning with a few labeled images. Wang et al. [12] solve the domain shift problem with a small amount of labeled target domain data by linearly modifying the parameters of the convolutional layers.

Although these semi-supervised methods reduce the amount

This work was supported in part by grants from the National Natural Science Foundation of China under Grant 61771348, and Tencent.

G. Ding and D. Yang are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (e-mail: gcding@whu.edu.cn, dqyang@whu.edu.cn).

T. Wang, S. Wang and Y. Zhang are with Tencent, Shenzhen 518000, China (email: tuckerwang@tencent.com; lovingwang@tencent.com; yanniszhang@tencent.com).

Corresponding author: Daiqin Yang.

of data that needs to be annotated, they still need manually labeled data for model training. In order to further ease the burden, some methods begin to use unsupervised learning [13]–[17]. Cycle GAN [18] is used in [13] to transfer the style of the GCC [13] data set to the target domain. CODA [14] combines adversarial learning and ranking to learn the density map of the target domain. FSC [15] pays attention to the semantic consistency of the source domain and target domain through PSPNet [19]. Although semantic alignment brings performance improvements, accurate semantic information is often extravagant. FA [16] designs a structured density map alignment to refine the quality of density maps and reduce domain gap in feature space. FineAdapt [17] proposes ASNet to gradually bridge the domain gap from coarse to fine so as to alleviate the generalization bottleneck caused by domain differences. However, these existing methods rarely pay attention to pixel-level or high-level feature alignment.

In this paper, we propose a novel cross-domain learning network for unsupervised domain adaptation. The proposed network consists of three parts, a Multi-granularity Feature-aware Discriminator (MFD) module, a Domain-invariant Feature Adaptation (DFA) module, and a Cross-domain Vanishing Bridge (CVB) module. The Multi-granularity Feature-aware Discriminator (MFD) module includes a Global Feature Discriminator (GFD) and a Local Feature Discriminator (LFD). GFD is used to discriminate whether the features extracted by the feature extractor network belong to the source domain or the target domain at image level. LFD is introduced to allow the network to distinguish the source domain and target domain more finely at local or even pixel levels. As the learning goal of MFD is to distinguish features from the two domains correctly, which is opposite to our goal of extracting domain invariant features for domain adaptation, a gradient reversal layer is inserted between the MFD module and the feature extract network to form an adversarial relationship and forces the feature extraction network to extract domain consistent features from the source domain and target domain. With extracted domain-invariant features, it is expected that these features should also be high-level aligned across the source and target domains. The Domain-invariant Feature Adaptation (DFA) module is thus proposed to further explicitly align the high-level marginal distributions of the features extracted from the source and target domains. As not the whole part of domain-invariant features are density map related, inspired by the Gradually Vanishing Bridge [20], the Cross-domain vanishing bridge (CVB) module is proposed to eliminate the non-density-map-related part of the extracted features from being mapped into the noise part of the density map.

The contributions of this paper are summarized as follows:

- A Multi-granularity Feature-aware Discriminator (MFD) module is proposed to focus on domain adaptation at both image level and pixel level, with the help of the GFD and LFD modules. The combined effect of these two modules empowers the feature extraction network both global and local granularity for the extraction of domain-invariant features.
- A Domain-invariant Feature Adaptation (DFA) module is proposed to further align the marginal distributions of the extracted features from the source and target domains. And a Cross-domain Vanishing Bridge (CVB) module is also proposed to use a bridge connection to remove the effect of non-density-map-related features.
- Comprehensive experiments are conducted and demonstrate that our cross-domain learning structure can achieve great performance improvement with various backbones, and the performance is better than or on par with the state-of-the-art methods.

II. Related work

In this section, recent learning-based crowd counting methods are reviewed first. Related topics on domain adaptation are then discussed.

A. Crowd counting

Due to the powerful performance of neural networks, learning-based algorithms have been widely used in crowd counting. The early method can be categorized as detection-and-count [21], which first detects people in the image and then count. However, it is difficult to detect people in a crowded scene accurately, and the bounding box annotation required for detection is a laborious process. [22], [23] avoid the use of detection but directly regresses through a feature vector to obtain the counting result. In these methods, the dot annotation maps are underutilized.

Lempitsky et al. [24] first convert the dot annotations into a ground truth density map using Gaussian kernels, and then train a density map estimator to regress the Gaussian density map for crowd counting. After that more and more algorithms [1], [5], [8], [13], [25]–[28] begin to use density map estimation. MCNN [8] proposes a three-column network with different kernel sizes to cover different receptive fields for the image. Switch-CNN [1] uses a switch to select the best one from multiple density map generators as the final result. Li et al. [25] propose a CNN model (CSRNet) that encodes more large-range features by a way of dilated convolution. SANet [26] proposes an efficient scale aggregation network to output structured density maps. SFCN [13] adds a spatial encoder to the top of the backbone so that the regression layer can obtain more information to generate the density map. Jiang et al. [5] use the attention mechanism and the pyramid loss function to propose ASNet to solve the scaling problem further. DensityCNN [29] uses a multi-task CNN structure to learn density level classification and density maps jointly. Liu et al. [30] propose DENet, which is composed of a DNNet for detecting and counting individuals and an ENNet for estimating the density map of the remaining area. PFDNet [31] models continuous scale variations and adaptively selects the proper dilation kernels to adapt to different spatial locations. AdaCrowd [32] proposes a guiding network that guides the crowd counting network to predict parameters based on the unlabeled images from a specific scene. Ma et al. [33] propose an SDNet to predict scale distributions of various scenes and address the catastrophic sensitivity of crowd counters by scale alignment.

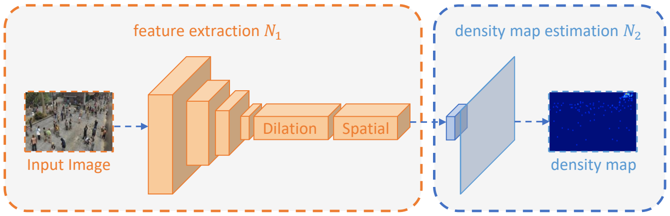


Fig. 2: General framework of counting using SFCN [13] as example.

Because of the high cost for data labeling, some algorithms [34]–[36] are devoted to crowd counting with scarce labeled data. Elassal et al. [34] first propose unsupervised crowd counting. Through 3D simulation, it can automatically understand how the crowd is displayed in the image for counting. Liu et al. [35] use ranked sub-image prior knowledge as a self-supervised condition for counting. Sam et al. [36] propose a near unsupervised method in which there is only 0.1% of the model parameters need labeled data for learning.

B. Domain adaptation

Traditional machine learning supposes that the training data and testing data are drawn i.i.d. from the same underlying distribution. However, this assumption is often not true in real-world, resulting in significant performance degradation across training and testing. Domain adaptation aims to reduce this mismatch so that the model can be better transferred to various scenarios.

Ganin et al. [37] propose DANN, which first introduces the idea of adversarial learning into domain adaptation. After that, domain adaptation is applied in many fields such as: object detection [38], [39], semantic segmentation [40]–[42] and image classification [20], [43]. Domain adaptation can be implemented into feature-level and image-level. Some algorithms [44]–[46] apply adversarial training at image level to make features invariant to illumination, color and style factors. These methods focus on converting the annotated data into synthetic data similar to the target data, while retaining the annotations. Yoo et al. [47] transfer knowledge from the source domain to pixel-level target images with GANs. Shrivastava et al. [48] propose a method of simulation plus unsupervised learning (S + U), which uses unlabeled real data to improve the authenticity of the synthesized image. Cui et al. [20] propose the gradually vanishing bridge mechanism on both generator and discriminator to obtain residual domain-specific characteristics. GDCAN [43] uses the attention mechanism to automatically explore low-level domain-specific features and feature adaptation blocks to reduce domain discrepancy.

Other algorithms [49]–[51] adopt adversarial training at feature level to learn domain-invariant features and reduce performance loss caused by domain-shifting. [14]–[16] all apply adversarial training at the feature level to force the network to learn domain-invariant features. Zhang et al. [52] propose a Crowd-Scene Crowd Counting (CSCC) method, which collects images similar to the target domain from the source domain, and then uses the selected images to fine-tune the pre-trained counting model. EDIREC-Net [53] transfers a pre-trained counting model to the target domain using an error-aware density isomorphism reconstruction objective and models the reconstruction erroneousness by error reasoning. Wu et al. [54] propose a zero-shot cross-domain crowd counting method C^{2}MoT, which dynamically updates pre-trained MoT for each test image through online cross-dataset evaluation. Liu et al. [55] propose a knowledge distillation method using an iterative self-supervised learning scheme to detect and count dual-source knowledge. [12] uses a semi-supervised learning method to extract domain-invariant features by linearly changing the parameters of the convolutional layer.

TABLE I: A list of notations mainly used in this paper.

Symbol	Definition
$\mathcal{D}, \mathcal{X}, \mathcal{G}$	Domain space, image space, annotation space and $\mathcal{D} = \{\mathcal{X}, \mathcal{G}\}$
B	batch size
H_i, W_i	Height and width of the corresponding image (feature)
N_1	Feature extraction network
N_2	Density estimation network
φ, ψ	mapping function
$\mathcal{L}_{sub}^{sup}$	The loss function of sub module in sup domain
d_i^s, d_i^t	Predicted density map from source domain and target domain
f_{sub}^{sup}	Features extracted by the sub module in the sup domain
x_i^s, x_i^t	Image from source domain and target domain

III. Framework and Approach

In this section, the general framework of the crowd counting network is depicted first. Then the approaches to achieve domain adaptation are described. After that, the detailed design of the cross-domain feature adaption network is introduced.

A. General framework of Counting

A domain $\mathcal{D}$ consists of an image space $\mathcal{X}$ and a point annotation (ground truth) space $\mathcal{G}$, where $X = \{x_1, x_2, \cdots, x_N\} \in \mathcal{X}$ and $G = \{g_1, g_2, \cdots, g_N\} \in \mathcal{G}$. Given a specific domain $\mathcal{D} = \{\mathcal{X}, \mathcal{G}\}$, CNN-based crowd counting methods aim at learning a mapping function $\varphi(x; \Theta)$ as the density map estimator with parameter set Θ. For each input image $x_i \in \mathbb{R}^{H_i \times W_i \times K}$, the density map $d_i \in \mathbb{R}^{H_i \times W_i}$ is estimated as:

$$d_i = \varphi(x_i; \Theta) \tag{1}$$

where H_i, W_i are the height and width of the image, K is the channel number of the image. The result for crowd counting is then calculated as:

$$result_{count} = \sum_{h=0}^{H_i} \sum_{w=0}^{W_i} d_i(h, w) \tag{2}$$

As shown in Fig. 2, the mapping function $\varphi(x_i; \Theta)$ can be decomposed into two networks: the feature extraction network N_1, and the density map estimation network N_2. The combination of network N_1 and network N_2 constitutes most of the existing CNN-based network paradigms, such as SFCN [13] and CSRNet [25].

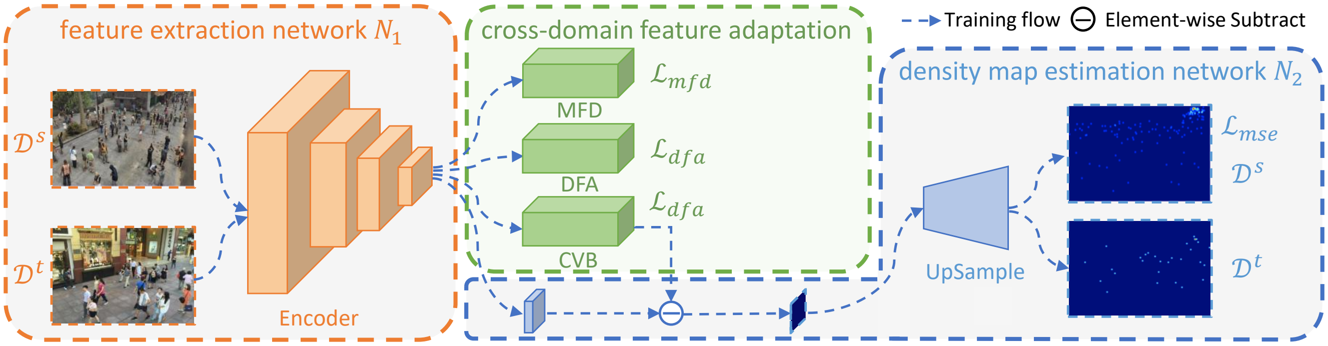


Fig. 3: Architecture of our cross-domain adaptation network. Detail of the MFD module and the DFA module are shown in Fig. 4 and Fig. 5, respectively. The meaning of the symbols in the figure is explained in detail in section III and table I.

Feature extraction network ***(N_1).*** Feature extraction is an important part of CNN-based crowd counting network. As the focus of this paper is to verify the effectiveness of the cross-domain learning network for domain adaptation, the design of the feature extraction network is not particularly important here. In the ablation study, several well-known networks are tested, including VGG16 [56], CSRNet [25], etc. The mapping from image x_i to feature f_i can be expressed by the formula:

$$f_i = \psi_f(x_i; \Theta_f) \tag{3}$$

Density estimation network ***(N_2).*** The density map estimation network is designed to transfer f_i into density map d_i. To keep computational and model complexity limited, N_2 is designed to simply convert the feature maps to a 1-channel feature map by a 3×3 convolution, and then output the density map through upsampling. The density map estimator can be expressed by the following formula:

$$d_i = \psi_d(f_i; \Theta_d) \tag{4}$$

Similar to previous work [13], [25], the standard MSE loss is used to train the feature extraction network N_1 and the density estimation network N_2 together. The input of the loss function is the predicted density map d_i and the ground truth density map generated by point annotations g_i. The formula is as follows:

$$\mathcal{L}_{mse} = \frac{1}{B}\sum_{i=1}^{B} |d_i - Gauss(g_i)|^2 \tag{5}$$

where $\mathrm{Gauss}(\cdot)$ is the Gaussian operation and B is the batch size for training. The same backbone of SFCN [13] as [13], [16], [57] is used to fairly compare our method with earlier algorithms. The SFCN network structure is illustration in Fig. 2.

B. Approaches for unsupervised domain adaptation

Assume there are two domains. One is the source domain with sufficient labeled point annotations, which is denoted as $\mathcal{D}^s = \{\mathcal{X}^s, \mathcal{G}^s\}$. The other is the target domain with no labeled information defined as $\mathcal{D}^t = \{\mathcal{X}^t, \varnothing\}$. Let's denote φ_s as the pre-trained model on $\mathcal{D}^s$ and $\varphi_{s,t}$ as the domain adapted model trained with both $\mathcal{D}^s$ and $\mathcal{D}^t$. Given a well-trained network φ_s, the objective of our proposed network is to learn a better $\varphi_{s,t}$ with the absence of $\mathcal{G}^t$ in $\mathcal{D}^t$.

In order to achieve the above objective, a new optimization goal is proposed to adapt the mapping function from φ_s to the target domain $\varphi_{s,t}$. In the re-training process for domain adaptation, a Multi-granularity Feature-aware Discriminator (MFD) module, a Domain-invariant Feature Adaptation (DFA) module and a Cross-domain Vanishing Bridge (CVB) module are introduced, in addition to the existing N_1 and N_2 network. With the absence of point annotation $\mathcal{G}^t$, MFD tries to utilize the simple category attributes $\mathcal{C}$ of each input image as a limited annotation for network training. $c_i = 1$ if the input image comes from $\mathcal{D}^s$ and $c_i = 0$ if the input image comes from $\mathcal{D}^t$. The purpose of MFD is to adjust the output features of N_1 to be representative for both $\mathcal{X}^s$ and $\mathcal{X}^t$ while keeping their image-level and pixel-level discriminative power for accurate density map estimation. DFA explicitly aligns the marginal distribution of the features extracted by the N_1 network through Maximum Mean Discrepancy (MMD) [58] loss to learn the domain invariant representation. The CVB module is inserted between N_1 and N_2 networks to further reduce the influence of non-density-map-related features in the output of N_1. The entire pipeline can be divided into three phases:

Phase1: Pre-train N_1 and N_2. As described in III-A, the general counting framework including N_1 and N_2 is first trained on the source domain.

Phase2: Re-train the entire network. The two networks, N_1, N_2, are re-trained with the help of the MFD, DFA and CVB modules, on both source and target domains. In this phase, no ground truth density maps are generated for target domain images. Neither Phase1 nor Phase2 is iterative, both of them are processed only once during training. Detailed network structures and loss functions are presented in the following subsections.

Phase3: Test the network. After the re-training phase for domain adaptation, testing images from the target domain are passed through the N_1 network, the CVB module and the N_2 network to obtain the estimated density map.

C. Multi-granularity Feature-aware Discriminator

The Multi-granularity Feature-aware Discriminator module is designed to help the feature extraction network N_1 fit the

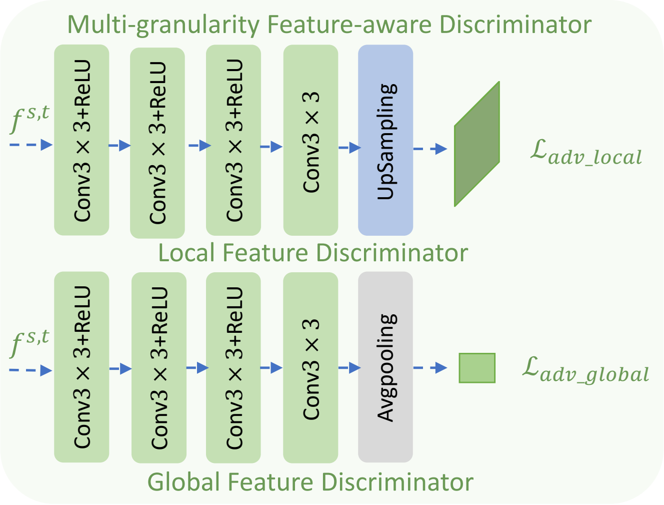

Multi-granularity Feature-aware Discriminator
$f^{s,t}$
Conv3 × 3+ReLU
Conv3 × 3+ReLU
Conv3 × 3+ReLU
Conv3 × 3
UpSampling
$\mathcal{L}_{adv\_local}$
Local Feature Discriminator
$f^{s,t}$
Conv3 × 3+ReLU
Conv3 × 3+ReLU
Conv3 × 3+ReLU
Conv3 × 3
Avgpooling
$\mathcal{L}_{adv\_global}$
Global Feature Discriminator

Fig. 4: Detail of the Multi-granularity Feature-aware Discriminator (MFD) module.

new domain $\mathcal{D}^t$ without the requirement of point annotation $\mathcal{G}^t$, which is usually expensive and time-consuming. In the re-training phase, the loss guided gradients propagated back from MFD are reversed in N_1. The reversion of gradient in N_1, makes N_1 and MFD an adversarial relationship. The training goal of N_1 becomes exactly the opposite of that of MFD. This plays an important role in the completion of domain adaptation. As the loss of MFD, $\mathcal{L}_{mfd}$, measures the extent of misclassification, it means that the extracted features of N_1 should prevent MFD from correctly classifying which domain these features come from. In other words, through the adversarial learning between MFD and N_1, the features extracted by N_1 will be transferred from the source domain feature space to the cross-domain feature space. At the same time, the loss of $\mathcal{L}_{mse}$, which is calculated on the source domain with ground truth of $\mathcal{G}^s$, will adapt the parameters of N_2 network to map the cross-domain features to accurate density map estimations.

Since crowd counting is a pixel-level regression task, the extracted features from N_1 should possess sufficient local features for N_2 to estimate an accurate density map. Meanwhile, the extracted features from N_1 should also contain adequate global features for accurate category classification. To meet these two different requirements through adversarial learning, as shown in Fig 4, the MFD module is designed to include both a Global Feature Discriminator module and a Local Feature Discriminator module.

Global Feature Discriminator(GFD): Like other domain adaptation methods [37], [59], [60], a global feature discriminator is used to determine whether the extracted features come from the source domain or the target domain. The GFD is composed of four convolutional layers with leaky ReLU and a fully connected layer. It converts the n-channel output feature f_i into a score value $\mathcal{S}_i$. The loss of the network is defined as the binary cross-entropy loss as follows:

$$\begin{aligned}\mathcal{L}_{adv\_global} = &-\frac{1}{B}\sum_{i=1}^{B}[(c_i \cdot \log(\mathcal{S}_i) \\ &+ (1-c_i)\cdot\log(1-\mathcal{S}_i)]\end{aligned} \tag{6}$$

where c_i is the category index about whether feature f_i comes from the source domain. With this loss function of GFD, the feature extract network N_1 will try its best to find a feature space that has the least distance between $\mathcal{D}^s$ and $\mathcal{D}^t$ while keeping them just distinguishable.

Local Feature Discriminator(LFD): Density map estimation is a pixel-by-pixel regression of the ground truth. It would be inadequate for the extracted feature of N_1 to only possess global similarity between $\mathcal{D}^s$ and $\mathcal{D}^t$ (at image distinguishable level). Pixel level common features among $\mathcal{D}^s$ and $\mathcal{D}^t$ should also be maintained for accurate density map estimation. To achieve this goal, the LFD module is proposed, which contains four convolutions with leaky ReLU. The input of LFD is f_i, and the output of LFD is a score map ($\mathcal{SM}$), in which the score of each pixel represents its probability of belonging to the source domain. Similar to GFD, binary cross-entropy loss, as defined below, is used to optimize LFD:

$$\begin{aligned}\mathcal{L}_{adv\_local} = &-\frac{1}{B\cdot H_i\cdot W_i}\sum_{i=1}^{B}\sum_{h=1}^{H_{f_i}}\sum_{w=1}^{W_{f_i}}[c_i \\ &\cdot\log(\mathcal{SM}_i(h,w)) + (1-c_i) \\ &\cdot\log(1-\mathcal{SM}_i(h,w))]\end{aligned} \tag{7}$$

where H_i and W_i are the height and width of feature f_i.

The loss of MFD is then defined as the sum of the two modules:

$$\mathcal{L}_{mfd} = \alpha\mathcal{L}_{adv\_global} + \beta\mathcal{L}_{adv\_local} \tag{8}$$

where α and β are set to 1 and 0.1 based on experience.

Compared with previous multi-discriminator domain adaptation networks [61], [62], MFD focuses on aligning features at multi-granularity (image granularity and pixel granularity) level because the density map based crowd counting is a pixel-wise regression task. However, the essence of the early algorithms [61], [62] is to capture multi-mode (i.e. different categories) structures and enable alignment of category distributions. We borrow the idea of multiple discriminators and propose two discriminators with two granularities, including a Global Feature Discriminator (GFD) and a Local Feature Discriminator (LFD), forming the framework of our MFD. From this perspective, our method is fundamentally different from the previous ones [61], [62].

D. Domain-invariant Feature Adaptation

It is difficult to extract pure domain invariant features only by relying on MFD due to unbalanced capability between N_1 and MFD. This imbalance stems from the following two reasons: 1. the discriminating ability of the MFD module is far less than the generating ability of N_1; 2. the N_1 network has been pre-trained on the source domain and the parameters of MFD are randomly initialized.

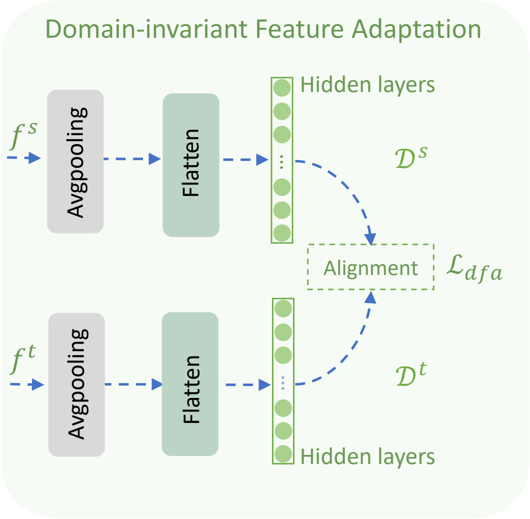


Fig. 5: Detail of the Domain-invariant Feature (DFA) module.

A DFA module is thus proposed to complement MFD. As shown in Fig 5, the DFA module tries to further align the distributions of features f^s and f^t, which are extracted by the N_1 network. To achieve this, the standard distribution distance metric MMD [58] is used to adjust the marginal distribution of the features. The classic MMD can be formulated as follows:

$$MMD(p,q) = \sup_{\|\psi\|_{\mathcal{H}} \leq 1} E_p(\psi(p)) - E_q(\psi(q)) \tag{9}$$

where E_p and E_q stand for mathematical expectations of variables p and q, and ψ is a non-linear mapping function that maps p, q into the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$. It has been proven that two distributions are equal if and only if $MMD(p,q) = 0$ [63].

The square of MMD can be calculated through kernel properties and the loss function of the DFA module is formulated as follows:

$$\begin{aligned}\mathcal{L}_{dfa} &= MMD^2(f^s_{dfa}, f^t_{dfa}) \\ &= \sup \|E[\psi(f^s_{dfa})] - E[\psi(f^t_{dfa})]\|^2 \\ &= E[K(f^s_{dfa}, f^s_{dfa})] + E[K(f^t_{dfa}, f^t_{dfa})] - \\ &\quad 2E[K(f^s_{dfa}, f^t_{dfa})]\end{aligned} \tag{10}$$

where f^s_{dfa} and f^t_{dfa} are the features after DFA module. $K(f^s_{dfa}, f^t_{dfa}) = \langle \psi(f^s_{dfa}), \psi(f^t_{dfa}) \rangle$ is the kernel function. In this paper, a Gaussian kernel is adopted.

The alignment capability of MMD loss conforms to the proposition that the transferability of features will decrease dramatically as the network [64] deepens. In DFA, the MMD loss is calculated for the extracted features, instead of the final output of estimated density maps in the previous algorithm of UDA [65]. A simple average pooling is applied in DFA to reduce the resolution of data before calculating the MMD.

E. Cross-domain Vanishing Bridge

For better comprehension of the CVB module, a toy example of domain adaptation effect is illustrated in Fig. 6. Let's define $\mathcal{F}^s$ and $\mathcal{F}^t$ as the source and target domain feature spaces that are extracted by N_1. $\mathcal{F}^i$ represents the density-map-related subset of the ideal domain invariant feature space among the source and target domains. With the training effect of MFD and DFA modules, N_1 tends to extract more domain invariant features. This effect is shown in the right side of Fig. 6, in which $\mathcal{F}^s$ and $\mathcal{F}^t$ move toward each other and their overlapping area enlarges a lot. As $\mathcal{F}^i$ is only the density-map-related portion of the whole domain invariant feature space, the overlapping area between $\mathcal{F}^s$ and $\mathcal{F}^t$ can exceed $\mathcal{F}^i$. The design purpose of CVB is to eliminate the impact of those domain specific, as well as non-density-map-related features (areas 1, 2 and 3 in Fig. 6) and prevent them from being mapped by N_2 to the noise part of the output density map.

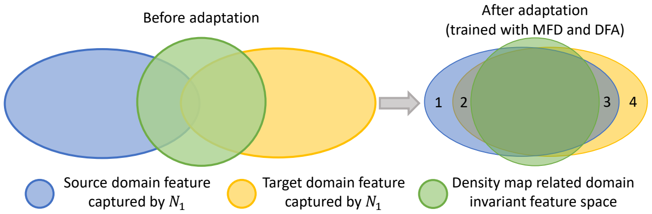

Before adaptation
After adaptation
(trained with MFD and DFA)
1
2
3
4
Source domain feature captured by N_1
Target domain feature captured by N_1
Density map related domain invariant feature space

Fig. 6: A toy example of the domain adaptation effect.

CVB module is a convolution layer with similar structure as in N_2. It is inserted into the N_2 network as shown in Fig. 3. The counter part of the original N_2 network is denoted as $conv$, and the newly inserted part in the CVB module is denoted as cvb. To remove the impact of non-density-map-related features, the CVB module is used as follows:

$$d_{i,\times 8} = \psi_{conv}(f_i) - \psi_{cvb}(f_i) \tag{11}$$

where ψ_{cvb} and ψ_{conv} represent the two convolutional mapping functions and $d_{i,\times 8}$ is the density map before upsampling.

During the training process for domain adaptation, domain-specific features contained in $f_i^{s,t}$ as well as their impact to the estimated density map are getting less and less. As the portion of non-density-map related domain invariant features is also small, the loss function is defined as follows:

$$\mathcal{L}_{cvb} = \frac{1}{(B_s + B_t) \cdot H_i \cdot W_i} \sum_{k=1}^{B_s+B_t} \sum_{w=1}^{W_i} \sum_{h=1}^{H_i} f^{s,t}_{cvb}(h,w) \tag{12}$$

Compared with the randomly initialized cvb, $conv$ in N_2 has a much stronger mapping ability with its well trained parameters. With the newly designed structure of N_2 and the loss function favoring zero output of cvb, when trained with the source domain data, cvb tends to capture the noisy part of the mapping result from non-density-map related features, including the mapping results from area 1, 2 and 3 in Fig. 6. Thus, although there is no labeled data in the target domain to tailor cvb with proper MSE loss, those training data from the source domain can enpower it with the ability to promote the performance in the target domain by removing the noisy mapping from those non-density-related domain-invariant features i.e. area 2 and 3 in Fig. 6.

IV. Implementation and Experiments

In this section, the implementation details are first introduced, including loss functions, parameter settings, data sets and evaluation metrics. Then, the results of our method migrating from real-world data set to real-world data set and from synthetic data set to real-world data set are reported. After that, our method is compared with state-of-the-art algorithms [1]. Finally, visualized results of our method are depicted to demonstrate the effectiveness of the algorithm.

A. Implementation details

1) Loss function: The loss functions in the two training phases are different:

Phase1: As described in section III-A, Eq. 5 is used as the loss function to pre-train the network on the source domain. This process is the same as most CNN-based methods.

Phase2: The loss functions of all the domain adaptation modules have been introduced in section III-C, III-D and III-E. For source domain training data, the loss function is defined as:

$$\mathcal{L}_s = \lambda_1 \mathcal{L}_{mfd} + \lambda_2 \mathcal{L}_{dfa} + \lambda_3 \mathcal{L}_{cvb} + \mathcal{L}_{mse} \tag{13}$$

For target domain training data, the loss function is defined as:

$$\mathcal{L}_t = \lambda_1 \mathcal{L}_{mfd} + \lambda_2 \mathcal{L}_{dfa} + \lambda_3 \mathcal{L}_{cvb} \tag{14}$$

2) Parameter settings and training strategy: As mentioned in section III, network training is mainly divided into two phases: 1) using source domain data for supervised learning, 2) using source domain data and target domain data for domain adaptation. For these two phases, the Adam [66] algorithm is used to optimize the network. The code is written under the C-3 framework [67], and NVIDIA 1080TI GPUs are used for network training and testing.

In the first phase, the learning rate is set to $e-5$. In the second phase, the learning rate for the density map estimation network and other parameters of the network are set to $e-5$ and $e-6$, respectively. The parameters are set to $\lambda_1 = e-3$, $\lambda_2 = e-5$ and $\lambda_3 = e-4$ according to a simple grid search using SHTA as source domain and SHTB as target domain.

3) Data sets: To evaluate the effectiveness of our method, we use a total of seven data sets GCC [13], Shanghai Tech A/B [8], UCF_QNRF [10], Mall [22], UCSD [68] and WorldExpo'10 [9] to conduct domain adaptation experiments.

GCC: GCC is short for GTA V Crowd Counting Data set, and all 15212 synthetic images are taken from 100 different locations in the GTA V game. The resolution of all images in this data set is 1080 × 1920.

Shanghai Tech Part A/B: Both Shanghai Tech Part A (SHTA) and Shanghai Tech Part B (SHTB) are randomly collected from the Internet. SHTA has a total of 482 images, with an average of 501 people per image, for a total of 241,677 heads. Its training set contains 300 images and 182 images are the test set. The resolution of SHTB is 768 × 1024, of which 400 images are the training set and 316 images are the test set following [8], a total of 88,488 head annotations.

UCF_QNRF: This data set contains 1,535 images with a total of 1,251,642 head annotations. Following [10], 1201 images are the training set, and the remaining 334 images are the test set. The average resolution of this data set is 2013 × 2902, and there are 815 people in each image on average. The free shooting perspective makes the data set more challenging.

Mall: This data set is obtained from a surveillance camera in a shopping mall. It contains 2000 images with resolution of 480 × 640. Following the [22], the first 800 images are training and the others are testing.

UCSD: This data set is collected on the sidewalk with a video camera. The data set contains 2000 frames with a low resolution of 158 × 238. Following the [68], 601 to 1400 images are training data and the others are testing data.

WorldExpo'10: This data set selects 1,132 video sequences from 108 surveillance cameras of the 2010 Shanghai World Expo event. The training set contains 103 cameras, and the rest are the testing set. To be specific, the data set has a total of 199,923 images with a resolution of 576 × 720, with an average of 50 people in each image.

The Matlab code provided by [69] is used to generate density maps. Because SHTA and UCF_QNRF data sets have different resolutions, their sizes are scaled to ensure that each image in SHTA is larger than 768 × 1024, and UCF_QNRF is larger than 1024 × 1024.

4) Evaluation metric: Two widely used metrics in crowd counting, Mean Absolute Error (MAE) and Mean Squared Error (MSE), are adopted to measure the performance of each model quantitatively. These two metrics are defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|d_i^{gt} - d_i\right| \tag{15}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|d_i^{gt} - d_i\right|^2} \tag{16}$$

where N is the number of testing images and d_i^{gt} is the ground truth of the i-th frame.

Based on MAE and MSE, the following formula is used to calculate the Performance Improvement Ratio (PIR) with domain adaptation:

$$PIR = (M_{no} - M_{da})/M_{no} \times 100\% \tag{17}$$

where M_{no} represents MAE/MSE before domain adaptation, and M_{da} represents the result after domain adaptation.

B. Real-World data set ⇒ Real-World data set

Our method is compared with state-of-the-art algorithms including CODA [14] and FA [16]. In this set of experiments, SHTA is used as the source domain to evaluate domain adaptation performances between real-world data sets. It is selected as the source domain for the following three reasons: 1. The average number of people per image in SHTA (501) is less than UCF_QNRF (815) and more than SHTB (123) and WorldExpo (50) data sets. It can verify the adaptation ability of our proposed network in terms of from simple data sets to crowded data sets, as well as from crowded data sets to simple data sets. 2. The images in the SHTA data set have different resolutions, and their average resolution is also

[1] Because most methods do not provide code, some results are obtained from public papers and quoted in the table.

TABLE II: Quantitative comparisons of SFCN based implementations with SHTA data set as the source domain. The best results are in red. Small MAE, small MSE and large PIR indicate good performance.

Method	Paradigm	SHTB		UCF_QNRF		WorldExpo'10
		MAE	MSE	MAE	MSE	MAE
NoAdpt [16]	NO	–	–	–	–	–
FA [16]	DA	14.8 ↓N/A	21.9 ↓N/A	–	–	–
NoAdpt [16]	NO	27.3	36.2	–	–	–
CODA [14]	DA	15.9 ↓ 41.76%	26.9 ↓ 25.69%	–	–	–
NoAdpt [55]	NO	–	–	–	–	–
RDBT [55]	DA	13.38 ↓N/A	29.25 ↓N/A	175.02 ↓N/A	294.76 ↓N/A	–
NoAdpt [17]	NO	27.28	35.14	–	–	–
CoarseAdapt [17]	DA	15.77 ↓ 42.19%	24.92 ↓ 29.08%	–	–	–
FineAdapt [17]	DA	13.59 ↓ 50.18%	23.15 ↓ 34.12%	–	–	–
NoAdapt	NO	20.89	31.72	164.87	309.92	25.17
ours	DA	13.16 ↓ 37.00%	23.42 ↓ 26.17%	137.67 ↓ 19.76%	253.92 ↓ 18.07%	21.09 ↓ 16.21%

TABLE III: Quantitative comparisons of BL based implementations with SHTA data set as the source domain. The best results are in red. Small MAE, small MSE and large PIR indicate good performance.

Method	SHTB		UCF_QNRF	
	MAE	MSE	MAE	MSE
NoAdpt [54]	15.9	25.8	166.7	287.6
C^{2}MoT (BL) [54]	12.4 ↓ 22.01%	21.1 ↓ 18.22%	125.7 ↓ 24.60%	218.3 ↓ 24.10%
ours (BL)	12.0 ↓ 24.53%	21.0 ↓ 18.60%	124.5 ↓ 25.31%	219.3 ↓ 23.75%

in the middle position (larger than WorldExpo'10, smaller than SHTB and UCF_QNRF), which is more representative. 3. In the literature that pays attention to domain adaptation between real-world data sets, most experiments are conducted with SHTA as the source domain. It is thus convenient for performance comparison.

As shown in Table II, when adapting to SHTB, our method achieves the best MAE as well as a competitive performance in MSE. When adapting to UCF_QNRF, our method achieves superior performance over RDBT [55] on both MAE and MSE. Our results of adapting to WorldExpo'10 are also reported in Table II. With domain adaptation, both MAE and MSE have varying degrees of performance improvements. Detailed results of the WorldExpo'10 data set are combined and reported in Table VI.

As C^{2}MoT [54] uses a stronger backbone of BL [70], a new version of our algorithm using the same backbone is implemented with the source code of BL [2]. Comparing results are listed in Table III. Using the pre-trained model on SHTA provided by BL, the same results are reproduced before domain adaptation. After domain adaptation, our method achieves the best MAE, MSE and PIR when adapting to SHTB. It also achieves comparable performances when adapting to UCF_QNRF. Compared with C^{2}MoT, which imposes more computational and memory consumptions over BL, such as dynamical updating of MoT, memory bank, etc., our method has almost the same inference speed as BL with just one more layer of convolution and pixel-wise subtraction.

C. Synthetic data set ⇒ Real-World data set

Our method is compared with state-of-the-art algorithms including CycleGAN [18], SE CycleGAN [13], SE CycleGAN (JT) [57], FSC [15], CODA [14] and FA [16]. Both the network compared and our SFCN [13] use VGG16 [56] as the backbone network. The GCC data set is the only synthetic data set among all the testing data sets, and it is used as the source domain in the experiment.

It is noteworthy that CycleGAN and FA give the results of domain adaptation using the GCC data set as the source domain. They both use Scene Regularization (SR) proposed by [13] on the GCC data set. SR selects those samples in the GCC data set which have similar weather, counting range, time and other factors as the target domain data set to form the source domain data set. In our experiments, when GCC data set is used as the source domain, we followed the same SR selection conditions as in [13], [16], and the largest subset of the filtered images is used as the source domain for domain adaptation. Density Regularization (DR) used in [13] sets an upper bound MAX_S to remove obvious errors in the predicted density map, and it is not adopted in our experiment.

1) Adaptation to Free view/scenes real-world data sets: Our proposed method is quantitatively compared with existing methods on the free view/scenes real-world data sets SHTA, SHTB, and UCF_QNRF. Although images in the SHTA data set are from the perspective of surveillance cameras, most of the images are taken by different cameras. So, SHTA is classified as a free scene.

As shown in Table IV, our method achieves the best performance in all the three data sets, in terms of MAE, MSE and PIR. On the SHTA and SHTB data sets, the performance of our network is inferior to comparison algorithms before domain adaptation. However, after applying domain adaptation, our algorithm not only has the largest performance improvement but also has the best performance of MAE and MSE. On the UCF_QNRF data set, before domain adaptation, the performance gap between our algorithm and the FSC algorithm is not significant. But after adaptation, our algorithm is far superior to FSC in MSE (PIR), and it is also better in

[2] https://github.com/ZhihengCV/Bayesian-Crowd-Counting

TABLE IV: Quantitative comparison of SFCN base implementations with GCC data set as the source domain. The best results are in red. Small MAE, small MSE and large PIR indicate good performance.

Method	Paradigm	Shanghai Tech Part A		Shanghai Tech Part B		UCF_QNRF	
		MAE	MSE	MAE	MSE	MAE	MSE
NoAdpt [57]	NO	160.0	216.5	22.8	30.6	275.5	458.5
CycleGAN [13]	DA	143.3 ↓ 10.44%	204.3 ↓ 5.64%	25.4 ↓ −11.40%	39.7 ↓ −29.74%	257.3 ↓ 6.61%	400.6 ↓ 12.63%
SE CycleGAN [13]	DA	123.4 ↓ 22.88%	193.4 ↓ 10.67%	19.9 ↓ 12.72%	28.3 ↓ 7.52%	230.4 ↓ 16.37%	384.5 ↓ 19.26%
SE Cycle GAN (JT) [57]	DA	119.6 ↓ 25.25%	189.1 ↓ 12.66%	16.4 ↓ 28.07%	25.8 ↓ 15.67%	225.9 ↓ 18.00%	385.7 ↓ 15.88%
NoAdpt [14]	NO	–	–	–	–	–	–
CODA [14]	DA	–	–	19.2 ↓ N/A	28.5 ↓ N/A	–	–
NoAdpt [15]	NO	190.8	298.1	24.6	33.7	296.1	467.9
FSC [15]	DA	129.3 ↓ 32.23%	187.6 ↓ 37.07%	16.9 ↓ 31.30%	24.7 ↓ 26.71%	221.2 ↓ 25.30%	390.2 ↓ 16.61%
NoAdpt [16]	NO	156.4	210.7	22.3	29.9	269.5	480.2
FA [16]	DA	144.6 ↓ 7.54%	200.6 ↓ 4.79%	16.0 ↓ 28.25%	24.7 ↓ 17.39%	255.4 ↓ 5.23%	407.9 ↓ 15.06%
NoAdapt	NO	225.28	310.81	29.82	44.26	287.86	467.33
ours	DA	116.50 ↓ 48.29%	182.18 ↓ 41.39%	15.90 ↓ 46.68%	24.08 ↓ 45.59%	207.96 ↓ 27.76%	333.66 ↓ 28.60%

TABLE V: Quantitative comparison of SFCN base implementations with GCC data set as the source domain. The best results are in red. Small MAE, small MSE and large PIR indicate good performance.

Method	Paradigm	Mall		UCSD		WorldExpo'10
		MAE	MSE	MAE	MSE	MAE
NoAdapt [57]	NO	–	–	–	–	42.8
CycleGAN [13]	DA	–	–	–	–	32.4 ↓ 24.30%
SE CycleGAN [13]	DA	–	–	–	–	26.3 ↓ 38.55%
SE Cycle GAN (JT) [57]	DA	–	–	–	–	24.4 ↓ 42.99%
NoAdapt [16]	NO	4.07	5.12	16.46	16.80	37.2
FA [16]	DA	2.47 ↓ 39.31%	3.25 ↓ 36.52%	2.00 ↓ 87.85%	2.43 ↓ 85.54%	21.6 ↓ 41.94%
NoAdapt	NO	13.85	14.23	22.40	23.27	61.07
ours	DA	2.56 ↓ 81.52%	3.18 ↓ 77.65%	4.30 ↓ 80.80%	6.12 ↓ 73.70%	28.32 ↓ 53.63%

TABLE VI: Quantitative comparison on the WorldExpo'10 in detail. Avg. represents the average of the previous five results. DR means density regularization. SR means the scene regularization. The best results are in red.

Method	Paradigm	DR	SR	Src data	WorldExpo'10 (MAE)					
					S1	S2	S3	S4	S5	Avg.
NoAdapt [13]	NO	✓	✓	GCC	4.4	87.2	59.1	51.8	11.7	42.8
CycleGAN [13]	DA	✓	✓	GCC	4.4	69.6	49.9	29.2	9.0	32.4 ↓ 24.30%
SE CycleGAN [13]	DA	✓	✓	GCC	4.3	59.1	43.7	17.0	7.6	26.3 ↓ 38.55%
SE CycleGAN (JT) [57]	DA	✓	✓	GCC	4.2	49.6	41.3	19.8	7.2	24.4 ↓ 42.99%
NoAdapt [16]	NO	×	✓	GCC	5.4	88.2	62.1	16.2	14.3	37.2
FA [16]	DA	×	✓	GCC	5.7	59.9	19.7	14.5	8.1	21.6 ↓ 41.94%
NoAdapt	NO	×	✓	GCC	14.67	110.15	79.72	79.28	21.55	61.07
ours	DA	×	✓	GCC	10.64	66.06	32.62	19.09	13.21	28.32 ↓ 53.63%
NoAdapt	NO	×	×	SHTA	5.99	55.85	40.34	15.70	7.97	25.17
ours	DA	×	×	SHTA	9.09	44.10	31.21	13.19	7.87	21.09 ↓ 16.21%

MAE (PIR). These experiments prove that our algorithm has great advantages in free view/scenes data sets.

2) Adaptation to Fixed view/scenes real-world data sets: As shown in Table V, the performance of our proposed method is compared with the SOTA methods on three fixed-view data sets Mall (1 Fixed Scene), UCSD (1 Fixed Scene), and WorldExpo'10 (108 Fixed Scenes). Although it is not as good as FA on UCSD, our proposed method achieves the greatest performance improvement on both Mall and WorldExpo'10. The resolution of the GCC (source domain) image is 1080×1920, and the resolution of the UCSD (target domain) image is 158×238, which is much lower than the source domain. The quality of UCSD images is also very low. As our method performs domain adaptation in feature space, which is further downsampled eight times related to the original image. It may suffer from severe noise with low resolution and low-quality images and general noisy results. This result shows that our method is more suitable for domain adaptations to high resolution and high-quality images.

Detailed results of the WorldExpo'10 data set are reported in Table VI and it could be found that our method performs the best when SHTA is the source domain. Although our

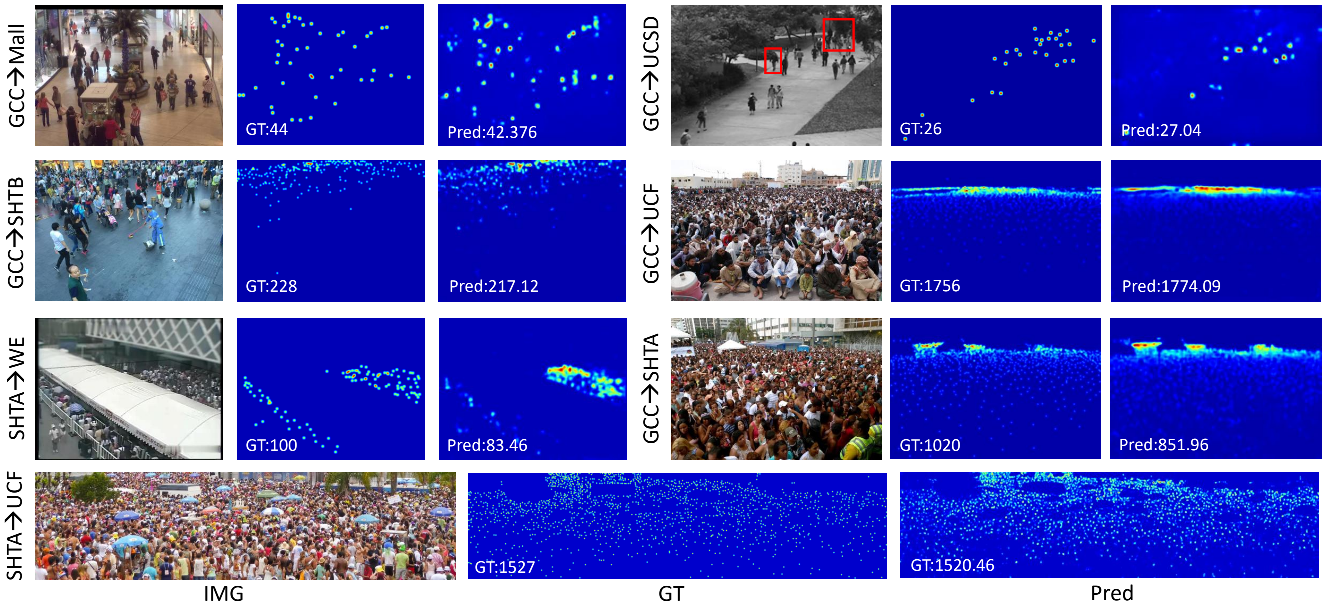


Fig. 7: Examples of visualization results on the WorldExpo'10, SHTA, UCF_QNRF, Mall and UCSD data sets. The number in the lower left corner of the image represents the number of people in the image.

performance is relatively poor when GCC is used as the source domain, it has the highest performance improvement, which is 10.64% higher than the second. It shows that our method has the strongest domain adaptability.

D. Qualitative comparison

Fig. 7 shows the visualized results of our method in five data sets: WorldExpo'10, SHTA, UCF_QNRF, Mall and UCSD. In general, our method can predict an effective density map, which can reflect the density of the crowd and the total number of people in the image. However, compared with the ground truth, the result is still a rough density map, which is affected by the background image. Especially for low-resolution images such as the UCSD data set, the predicted density map does not distinguish people well from the background. But this phenomenon does not appear in other data sets. The reason for this phenomenon is that people and the surrounding environment have similar textures (such as the people shown in the red frames of Fig. 7) in low-resolution and low-quality images.

V. Ablation Study

In this section, ablation experiments are conducted and the experiment results are discussed to evaluate the contribution of different modules and the effectiveness of our proposed adaption structure and different modules.

A. Robustness to different backbones

In order to prove the effectiveness of the proposed network, different network structures are used as N_1 and N_2 for comparative experiments. There are two types of networks for the ablation experiments. The first type is a general CNN-based network represented by VGG16 [56] and ResNet50 [71]. For these two networks, they are used as the feature extraction part of N_1 followed by a layer of convolution and an upsampling layer to directly regression to get a density map. The second type is a network proposed explicitly for crowd counting represented by CSRNet [25] and SFCN [13]. These two networks are chosen because of their wide adoption in cross-domain crowd counting approaches.

TABLE VII: Quantitative evaluation with different network backbones. The red numbers and blue numbers represent the first and second best performers, respectively.

Backbone	NoAdapt		Adapted (SHTA → SHTB)	
	MAE	MSE	MAE	MSE
VGG16	24.55	39.64	15.67 ↓ 36.17%	26.87 ↓ 32.21%
ResNet50	17.93	31.41	12.34 ↓ 28.50%	21.36 ↓ 28.69%
CSRNet	26.77	38.30	13.85 ↓ 48.26%	24.85 ↓ 35.12%
SFCN	20.89	31.72	13.16 ↓ 37.00%	23.42 ↓ 26.17%

The experimental results are shown in Table VII. In the two metrics of MAE and MSE, different degrees of performance improvement can be achieved after domain adaptation. Take SFCN as an example, when the model pre-trained on SHTA is directly used to test on SHTB, MAE and MSE are 20.89 and 31.72, respectively. After using the proposed network for domain adaptation, MAE and MSE are reduced by 37.00% and 26.17%, respectively. Compared with the other three networks with VGG as a backbone, ResNet50 has the best performance before and after domain adaptation. This could be attributed to that the ResNet50 network has more robust feature extraction capabilities than the VGG network, and deeper features contain less domain information and perform better in generalization. In the two metrics of MAE and MSE, even the ResNet50 network can improve 28.50% and 28.69%, respectively. This also verifies to a certain extent that the

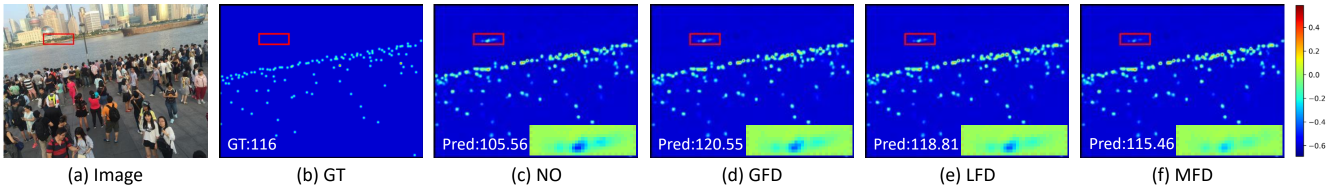


Fig. 8: Visualized results of the MFD module. The schematic diagram in the lower right corner is the difference between the estimated density map and the ground truth density map of the range corresponding to the red rectangle. The color bar corresponding to the image in the lower right corner is on the right. The same settings are used in Fig. 9 and Fig. 10.

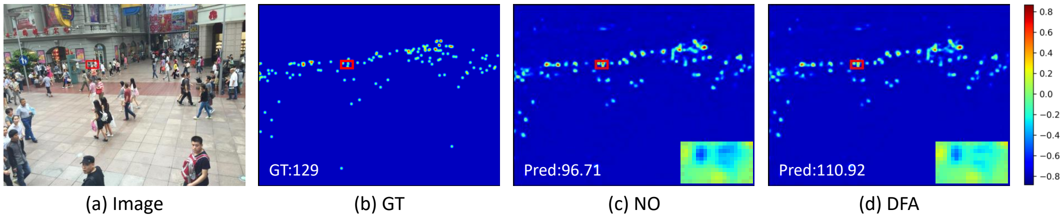


Fig. 9: Quantitative comparison of DFA module.

proposed network has strong cross-domain feature adaptation capabilities.

B. Contribution of MFD

To examine the effectiveness of the MFD module in the proposed network, the network is re-trained with Global Feature Discriminator only, Local Feature Discriminator only, and the combined Multi-granularity Feature-aware Discriminator, respectively. As shown in Table VIII, compared with the method that does not use domain adaptation, the network that uses only GFD or LFD has an improvement of 31.26% and 23.20% in terms of MAE, 31.97% and 22.76% in terms of MSE, respectively. The network using the combined MFD module has an improvement of 33.57% and 23.80% on MAE and MSE, respectively, which are better than just using GFD or LFD only. This result shows that joint constraints at the image level and pixel level can improve the adaptation ability of the network.

Fig. 8 gives a closer look at the differences between the predicted density map and the ground truth density map (the enlarged heat map shows the difference between them). A region with no people but a confusing background is highlighted and enlarged at the lower right corner of each predicted density map. From the zoomed-in image, it could be found that the density map predicted by GFD, LFD and MFD can better respond to the complex background, especially the MFD. Errors caused by complex backgrounds are effectively reduced. This result shows that the MFD module combines the gain from GFD and LFD and improves the network's ability to understand complex scenes by focusing on both image level and pixel level.

C. Contribution of DFA

In order to prove that the DFA module can reduce the domain gap by aligning the marginal distribution of the extracted features, a network is trained with only the DFA module. It can be seen from Table VIII that using DFA only, the two evaluation metrics, MAE and MSE, have increased by 33.57% and 23.87%, respectively, compared to without domain adaptation.

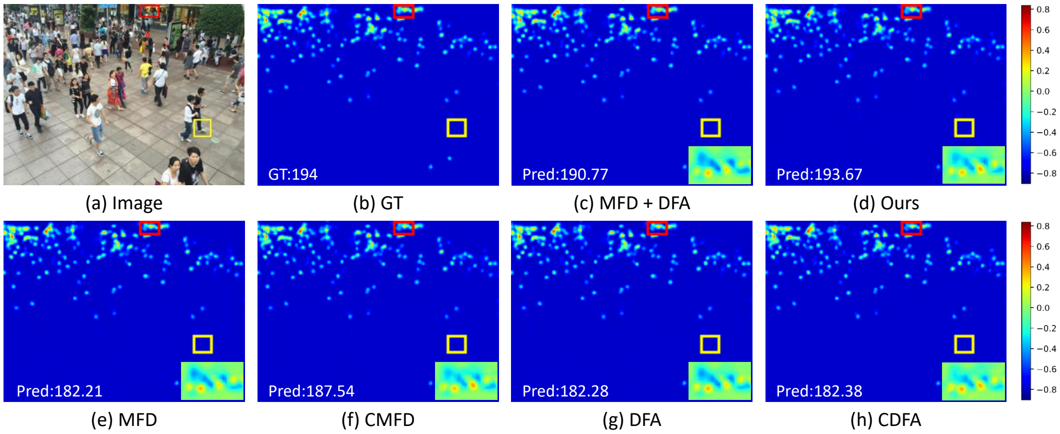
TABLE VIII: Quantitative evaluation of different domain adaptation modules. The red numbers and blue numbers represent the first and second best performers, respectively.

Method	SHTA ⇒ SHTB	
	MAE	MSE
NoAdapt(SFCN)	20.89	31.72
GFD	14.36 ↓ 31.26%	24.36 ↓ 23.20%
LFD	14.21 ↓ 31.97%	24.50 ↓ 22.76%
MFD	13.88 ↓ 33.57%	24.17 ↓ 23.80%
MFD + CVB	13.41 ↓ 35.81%	24.08 ↓ 24.09%
DFA	13.88 ↓ 33.57%	24.15 ↓ 23.87%
DFA + CVB	13.71 ↓ 34.37%	23.77 ↓ 25.06%
MFD + DFA	13.37 ↓ 36.00%	23.63 ↓ 25.50%
MFD + DFA + CVB	13.16 ↓ 37.00%	23.42 ↓ 26.17%

A visualized demonstration is given in Fig. 9. The highlighted region is a remote part of the scene with people mixed with the background. It can be found that by aligning high-level domain-general features with the DFA module, the proposed network can better adapt to challenging scenarios in the target domain to obtain a more accurate density map.

D. Contribution of CVB

The CVB module is designed to remove the impact of non-density-map-related features. We use MFD + CVB, DFA + CVB and MFD + DFA + CVB to train the network and compare with MFD, DFA and MFD + DFA respectively. The experimental results shown in Table VIII show that after adding the CVB module, MFD experiences a further drop of 2.24% and 0.29% in the two metrics of MAE and MSE, DFA experiences a further drop of 0.80% and 1.19%, and MFD + DFA experiences a further drop of 1.00% and 0.67%.



(a) Image (b) GT (c) MFD + DFA (d) Ours

(e) MFD (f) CMFD (g) DFA (h) CDFA

Fig. 10: Visualization of the CVB module. The yellow rectangle encloses the area that is easy to be confused.

Fig. 10 shows the visualized effect of the CVB module. It can be seen from the zoomed-in image that after adding the CVB module to MFD, DFA and MFD + DFA, the network can predict more accurately for crowded places in the distance (compared to the without case).

In Fig. 10, the yellow rectangles highlight a random part of the backgrounds (4900 pixels, with 70 pixels in length and width), which is difficult for us to see any difference with naked eyes. The counting errors between the predicted density maps and the GT density map in these yellow rectangles are $2.71e-4$, $5.74e-5$, $1.40e-2$, $1.85e-4$, $1.43e-2$ and $2.46e-4$, for MFD + DFA, MFD + DFA +CVB, MFD, CMFD, DFA and CDFA, respectively. It also proves that the CVB module can remove the impact of non-density-map-related features and provide the network with better mapping performance, especially in areas with confusing backgrounds.

VI. Conclusion

In this paper, we present an unsupervised cross-domain feature adaptation network to achieve domain-invariant features by simultaneously learning in low-level feature space and aligning feature distribution at higher levels. Different from previous domain adaptation work, through the MFD module, the network can not only align features from image level but also refine features from pixel level. The DFA module, in the network can promote high-level domain-invariant feature extractions by aligning the distributions of the features extracted from the two domains. Besides these, with the CVB module the impacts of those residual non-density-map-related features are removed through a bridge structure. The performance of our network is evaluated with two types of domain adaptation experiments on seven public data sets, and the proposed network presents competitive performance to the state-of-the-art methods.

References

[1] D. B. Sam, S. Surya, and R. V. Babu, “Switching convolutional neural network for crowd counting,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 4031–4039, doi:10.1109/CVPR.2017.429. 1, 2

[2] A. Zhang, J. Shen, Z. Xiao, F. Zhu, X. Zhen, X. Cao, and L. Shao, “Relational attention network for crowd counting,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6788–6797, doi:10.1007/978-3-319-46478-7_38. 1

[3] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, and H. Wu, “ADCrowdNet: An attention-injective deformable convolutional network for crowd understanding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3225–3234, doi:10.1109/CVPR.2019.00334. 1

[4] D. Guo, K. Li, Z.-J. Zha, and M. Wang, “DADNet: Dilated-attention-deformable convnet for crowd counting,” in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1823–1832, doi:10.1145/3343031.3350881. 1

[5] X. Jiang, L. Zhang, M. Xu, T. Zhang, P. Lv, B. Zhou, X. Yang, and Y. Pang, “Attention scaling for crowd counting,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4706–4715, doi:10.1109/CVPR42600.2020.00476. 1, 2

[6] J. Cheng, H. Xiong, Z. Cao, and H. Lu, “Decoupled two-stage crowd counting and beyond,” *IEEE Transactions on Image Processing*, vol. 30, pp. 2862–2875, 2021, doi:10.1109/TIP.2021.3055631. 1

[7] J. Wan, N. S. Kumar, and A. B. Chan, “Fine-grained crowd counting,” *IEEE Transactions on Image Processing*, vol. 30, pp. 2114–2126, 2021, doi:10.1109/TIP.2021.3049938. 1

[8] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, “Single-image crowd counting via multi-column convolutional neural network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 589–597, doi:10.1109/CVPR.2016.70. 1, 2, 7

[9] C. Zhang, K. Kang, H. Li, X. Wang, R. Xie, and X. Yang, “Data-driven crowd understanding: A baseline for a large-scale crowd dataset,” *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1048–1061, 2016, doi:10.1109/TMM.2016.2542585. 1, 7

[10] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah, “Composition loss for counting, density map estimation and localization in dense crowds,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 532–546, doi:10.1007/978-3-030-01216-8_33. 1, 7

[11] M. K. K. Reddy, M. Hossain, M. Rochan, and Y. Wang, “Few-shot scene adaptive crowd counting using meta-learning,” in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2814–2823, doi:10.1109/WACV45572.2020.9093409. 1

[12] Q. Wang, T. Han, J. Gao, and Y. Yuan, “Neuron linear transformation: Modeling the domain shift for crowd counting,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2021, doi:10.1109/TNNLS.2021.3051371. 1, 3

[13] Q. Wang, J. Gao, W. Lin, and Y. Yuan, “Learning from synthetic data for crowd counting in the wild,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8198–8207, doi:10.1109/CVPR.2019.00839. 2, 3, 4, 7, 8, 9, 10

[14] L. Wang, Y. Li, and X. Xue, “CODA: counting objects via scale-aware adversarial density adaption,” in *IEEE International Conference on Multimedia and Expo, ICME 2019, Shanghai, China, July 8-12, 2019*. IEEE, 2019, pp. 193–198, doi:10.1109/ICME.2019.00041. 2, 3, 7, 8, 9

[15] T. Han, J. Gao, Y. Yuan, and Q. Wang, “Focus on semantic consistency for cross-domain crowd understanding,” in *Proceedings of the IEEE*

International Conference on Acoustics, Speech and Signal Processing. IEEE, 2020, pp. 1848–1852, doi:10.1109/ICASSP40776.2020.9054768. 2, 3, 8, 9

[16] J. Gao, Q. Wang *et al.*, "Feature-aware adaptation and density alignment for crowd counting in video surveillance," *IEEE Transactions on Cybernetics*, 2020, doi:10.1109/TCYB.2020.3034316. 2, 3, 4, 7, 8, 9

[17] Z. Zou, X. Qu, P. Zhou, S. Xu, X. Ye, W. Wu, and J. Ye, *Coarse to Fine: Domain Adaptive Crowd Counting via Adversarial Scoring Network*. New York, NY, USA: Association for Computing Machinery, 2021, p. 2185–2194, doi:10.1145/3474085.3475377. 2, 8

[18] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232, doi:10.1109/ICCV.2017.244. 2, 8

[19] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2881–2890, doi:10.1109/CVPR.2017.660. 2

[20] S. Cui, S. Wang, J. Zhuo, C. Su, Q. Huang, and Q. Tian, "Gradually vanishing bridge for adversarial domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 455–12 464, doi:10.1109/CVPR42600.2020.01247. 2, 3

[21] W. Ge and R. T. Collins, "Marked point processes for crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2913–2920, doi:10.1109/CVPR.2009.5206621. 2

[22] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting." in *Proceedings of the IEEE British Machine Vision Conference*, vol. 1, no. 2, 2012, p. 3, doi:10.5244/C.26.21. 2, 7

[23] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015, pp. 1299–1302, doi:10.1145/2733373.2806337. 2

[24] V. Lempitsky and A. Zisserman, "Learning to count objects in images," *Advances in Neural Information Processing Systems*, vol. 23, pp. 1324–1332, 2010. [Online]. Available: https://proceedings.neurips.cc/paper/2010/file/fe73f687e5bc5280214e0486b273a5f9-Paper.pdf 2

[25] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1091–1100, doi:10.1109/CVPR.2018.00120. 2, 3, 4, 10

[26] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 734–750, doi:10.1007/978-3-030-01228-1_45. 2

[27] S. Bai, Z. He, Y. Qiao, H. Hu, W. Wu, and J. Yan, "Adaptive dilated network with self-correction supervision for counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4594–4603, doi:10.1109/CVPR42600.2020.00465. 2

[28] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6142–6151, doi:10.1109/ICCV.2019.00624. 2

[29] X. Jiang, L. Zhang, T. Zhang, P. Lv, B. Zhou, Y. Pang, M. Xu, and C. Xu, "Density-aware multi-task learning for crowd counting," *IEEE Transactions on Multimedia*, vol. 23, pp. 443–453, 2021, doi:10.1109/TMM.2020.2980945. 2

[30] L. Liu, J. Jiang, W. Jia, S. Amirgholipour, Y. Wang, M. Zeibots, and X. He, "DENet: A universal network for counting crowd with varying densities and scales," *IEEE Transactions on Multimedia*, vol. 23, pp. 1060–1068, 2020, doi:10.1109/TMM.2020.2992979. 2

[31] Z. Yan, R. Zhang, H. Zhang, Q. Zhang, and W. Zuo, "Crowd counting via perspective-guided fractional-dilation convolution," *IEEE Transactions on Multimedia*, pp. 1–1, 2021, doi:10.1109/TMM.2021.3086709. 2

[32] M. K. Krishnareddy, M. Rochan, Y. Lu, and Y. Wang, "AdaCrowd: Unlabeled scene adaptation for crowd counting," *IEEE Transactions on Multimedia*, pp. 1–1, 2021, doi:10.1109/TMM.2021.3062481. 2

[33] Z. Ma, X. Hong, X. Wei, Y. Qiu, and Y. Gong, "Towards a universal model for cross-dataset crowd counting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 3205–3214, doi:10.1109/ICCV48922.2021.00319. 2

[34] N. Elassal and J. H. Elder, "Unsupervised crowd counting," in *Proceedings of the IEEE Asian Conference on Computer Vision*. Springer, 2016, pp. 329–345, doi:10.1007/978-3-319-54193-8_21. 3

[35] X. Liu, J. Van De Weijer, and A. D. Bagdanov, "Leveraging unlabeled data for crowd counting by learning to rank," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7661–7669, doi:10.1109/CVPR.2018.00799. 3

[36] D. B. Sam, N. N. Sajjan, H. Maurya, and R. V. Babu, "Almost unsupervised learning for dense crowd counting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8868–8875, doi:10.1609/aaai.v33i01.33018868. 3

[37] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016. [Online]. Available: https://jmlr.org/papers/volume17/15-239/15-239.pdf 3, 5

[38] C.-D. Xu, X.-R. Zhao, X. Jin, and X.-S. Wei, "Exploring categorical regularization for domain adaptive object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 724–11 733, doi:10.1109/CVPR42600.2020.01174. 3

[39] M. Xu, H. Wang, B. Ni, Q. Tian, and W. Zhang, "Cross-domain detection via graph-induced prototype alignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 355–12 364, doi:10.1109/CVPR42600.2020.01237. 3

[40] G. Li, G. Kang, W. Liu, Y. Wei, and Y. Yang, "Content-consistent matching for domain adaptive semantic segmentation," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 440–456, doi:10.1007/978-3-030-58568-6_26. 3

[41] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang, "Adversarial style mining for one-shot unsupervised domain adaptation," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 20 612–20 623. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/ed265bc903a5a097f61d3ec064d96d2e-Paper.pdf 3

[42] Q. Lian, F. Lv, L. Duan, and B. Gong, "Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6758–6767, doi:10.1109/ICCV.2019.00686. 3

[43] S. Li, B. Xie, Q. Lin, C. H. Liu, G. Huang, and G. Wang, "Generalized domain conditioned adaptation network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, doi:10.1109/TPAMI.2021.3062644. 3

[44] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "CrDoCo: Pixel-level domain transfer with cross-domain consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1791–1800, doi:10.1109/CVPR.2019.00189. 3

[45] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 1989–1998. [Online]. Available: https://proceedings.mlr.press/v80/hoffman18a.html 3

[46] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6936–6945, doi:10.1109/CVPR.2019.00710. 3

[47] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon, "Pixel-level domain transfer," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 517–532, doi:10.1007/978-3-319-46484-8_31. 3

[48] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2107–2116, doi:10.1109/CVPR.2017.241. 3

[49] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176, doi:10.1109/CVPR.2017.316. 3

[50] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017, pp. 6670–6680. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/21c5bba1dd6aed9ab48c2b34c1a0adde-Paper.pdf 3

[51] R. Volpi, P. Morerio, S. Savarese, and V. Murino, "Adversarial feature augmentation for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5495–5504, doi:10.1109/CVPR.2019.00710. 3

[52] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 833–841, doi:10.1109/CVPR.2015.7298684. 3

[53] Y. He, Z. Ma, X. Wei, X. Hong, W. Ke, and Y. Gong, "Error-aware density isomorphism reconstruction for unsupervised cross-domain crowd counting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1540–1548. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16245 3

[54] Q. Wu, J. Wan, and A. B. Chan, *Dynamic Momentum Adaptation for Zero-Shot Cross-Domain Crowd Counting*. New York, NY, USA: Association for Computing Machinery, 2021, p. 658–666, doi:10.1145/3474085.3475230. 3, 8

[55] Y. Liu, Z. Wang, M. Shi, S. Satoh, Q. Zhao, and H. Yang, *Towards Unsupervised Crowd Counting via Regression-Detection Bi-Knowledge Transfer*. New York, NY, USA: Association for Computing Machinery, 2020, p. 129–137, doi:10.1145/3394171.3413825. 3, 8

[56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556 4, 8, 10

[57] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Pixel-wise crowd understanding via synthetic data," *International Journal of Computer Vision*, vol. 129, no. 1, pp. 225–245, 2021, doi:10.1007/s11263-020-01365-4. 4, 8, 9

[58] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two-sample-problem," *Advances in Neural Information Processing Systems*, vol. 19, pp. 513–520, 2006. [Online]. Available: https://proceedings.neurips.cc/paper/2006/file/e9fb2eda3d9c55a0d89c98d6c54b5b3e-Paper.pdf 4, 6

[59] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "FCNs in the Wild: Pixel-level adversarial and constraint-based adaptation," *arXiv preprint arXiv:1612.02649*, 2016. [Online]. Available: http://arxiv.org/abs/1612.02649 5

[60] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7472–7481, doi:10.1109/CVPR.2018.00780. 5

[61] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 3934–3941, Apr. 2018. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17067 5

[62] L. Hu, M. Kan, S. Shan, and X. Chen, "Unsupervised domain adaptation with hierarchical gradient synchronization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, doi:10.1109/CVPR42600.2020.00410. 5

[63] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2188410 6

[64] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014, pp. 3320–3328. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/375c71349b295fbe2dcdca9206f20a06-Paper.pdf 6

[65] M. A. Hossain, M. K. K. Reddy, K. Cannons, Z. Xu, and Y. Wang, "Domain adaptation in crowd counting," in *2020 17th Conference on Computer and Robot Vision (CRV)*, 2020, pp. 150–157, doi:10.1109/CRV50864.2020.00028. 6

[66] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980 7

[67] J. Gao, W. Lin, B. Zhao, D. Wang, C. Gao, and J. Wen, "C^3 framework: An open-source pytorch code for crowd counting," *arXiv preprint arXiv:1907.02724*, 2019. [Online]. Available: http://arxiv.org/abs/1907.02724 7

[68] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–7, doi:10.1109/CVPR.2008.4587569. 7

[69] Q. Wang, J. Gao, W. Lin, and X. Li, "NWPU-Crowd: A large-scale benchmark for crowd counting and localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, doi:10.1109/TPAMI.2020.3013269. 7

[70] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 6141–6150, doi:10.1109/ICCV.2019.00624. 8

[71] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778, doi:10.1109/CVPR.2016.90. 10