Neuron Linear Transformation: Modeling the Domain Shift for Crowd Counting

Qi Wang, *Senior Member, IEEE*, Tao Han, *Student Member, IEEE*, Junyu Gao, *Member, IEEE*, and Yuan Yuan, *Senior Member, IEEE*

***Abstract*—Cross-domain crowd counting (CDCC) is a hot topic due to its importance in public safety. The purpose of CDCC is to alleviate the domain shift between the source and target domain. Recently, typical methods attempt to extract domain-invariant features via image translation and adversarial learning. When it comes to specific tasks, we find that the domain shifts are reflected on model parameters' differences. To describe the domain gap directly at the parameter-level, we propose a Neuron Linear Transformation (NLT) method, exploiting domain factor and bias weights to learn the domain shift. Specifically, for a specific neuron of a source model, NLT exploits few labeled target data to learn domain shift parameters. Finally, the target neuron is generated via a linear transformation. Extensive experiments and analysis on six real-world datasets validate that NLT achieves top performance compared with other domain adaptation methods. An ablation study also shows that the NLT is robust and more effective than supervised and fine-tune training. Code is available at: https://github.com/taohan10200/NLT.**

***Index Terms*—Neuron linear transformation, crowd counting, domain adaptation, few-shot learning**

I. Introduction

Currently, accelerating the crowd understanding is playing an increasingly important role in building an intelligent society. As a huge research field, it involves many hotspots. In some scenes with sparse crowd distribution, crowd understanding mainly includes crowd detection [1], groups analysis [2], crowd segmentation [3], and crowd tracking [4]. In some scenes with dense crowds, such as an image containing thousands of people, crowd understanding mainly focuses on counting and density estimation [5], [6], [7], [8], [9]. In this paper, we strive to work on the existing crowd counting problem.

Crowd counting, a system that generates a pixel-level density estimation map and sums all of the pixels to predict how many people are in an image, has become a prevalent task due to its widespread piratical application: public management, traffic flow prediction, scene understanding [10], video analysis [11], *etc*. Specifically, it can be used for public safety in many situations, such as political rallies and sports events [12]. Besides, density estimation can also help crowd localization in some sparse scenes [13]. In the traditional supervised learning, many excellent algorithms [6], [14], [15], [16], [17] constantly refresh the counting metrics from different angles for the existing datasets.

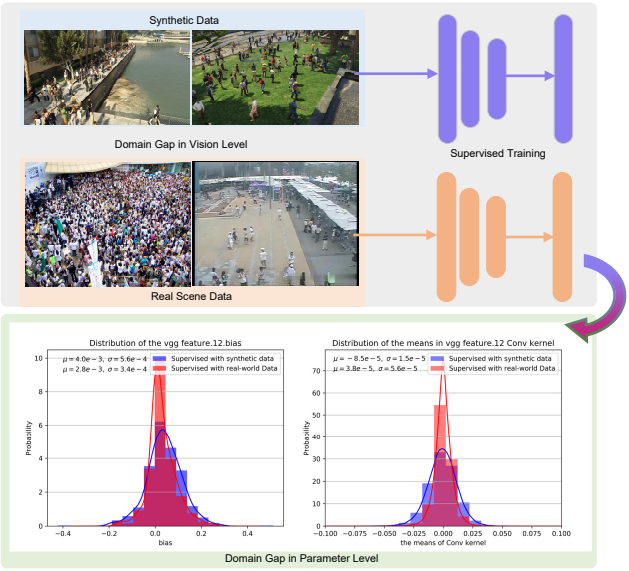


Fig. 1: The domain shift in different views. 1) visual domain shift, such as brightness, background, character feature, etc. 2) when it comes to specific tasks, the domain shift is reflected in the model's parameter distribution.

However, traditional supervised learning requires a lot of labeled data to drive it, and unfortunately, pixel-level annotating is often costly. According to statistics [13], the entire procedure involves 2, 000 human-hours spent in completing the QNRF dataset [13]. For the recently established NWPU dataset [18], the time cost is even as high as 3,000 human-hours. Even if researchers devote a lot of time and money to construct the datasets, the existing datasets are still limited in scale.

Because of the small-scale in some existing datasets, the above models may suffer from overfitting at different extents. It causes a significant performance reduction when applying them in real life. Thus, CDCC attracts the researcher's attention, which focuses on improving the performance in the target domain by using the data from the source domain. Wang *et al.* [19] propose a crowd counting via domain adaptation method, SE CycleGAN, which translates synthetic data to photo-realistic scenes, and then apply the trained model in the

Manuscript received March 16, 2020; first revised September 16, 2020; second revised December 19, 2020; accepted January 08, 2021. This work was supported by the National Natural Science Foundation of China under Grant U1864204, 61773316, 61632018, and 61825603.

The authors are with the School of Computer Science and the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: crabwq@gmail.com; hantao10200@mail.nwpu.edu.cn, gjy3035@gmail.com; y.yuan@nwpu.edu.cn). Yuan Yuan is the corresponding author.

wild. Gao *et al.* [20] present a high-quality image translation method feature disentanglement. [21], [22] adopt the adversarial learning to extract the domain-invariant features in the source and target domain. In a word, general Unsupervised Domain Adaption (UDA) methods concentrate on image style and feature similarity. The upper box in Fig. 1 demonstrates the appearance differences.

Nevertheless, the domain shift in image and feature level is not sensitive to the counting task: this strategy does not directly affect the counting performance, and it is not optimal. For example, SE CyCleGAN [19], and DACC [20] focus on maintaining the local consistency to improve the translation quality in congested regions. When applying the model to the sparse scenes (Mall [23], UCSD [24]), the loss may be redundant. In other words, there are task gaps in the existing UDA-style methods. Besides, since the target label is unseen for UDA models, they do not work well, such as coarse prediction in the congested region and the estimation errors in the background.

Given a specific task, we find that the parameters' difference between two models can reflect the domain shift. Notably, we use synthetic data and real-scene data to train the model, respectively. Then, we calculate the mean value of each kernel in a specific layer. The bottom box in Fig. 1 reports the distribution histogram. It can intuitively see that the parameters both show Gaussian distribution, and the differences are their mean and variance. Thus, we assume that the difference in parameter distribution can be exploited to measure the domain shift in two datasets.

According to the above observation, the domain shift on the parameter level can be simulated by a linear transformation. Thus, this paper proposes an NLT method to tackle cross-domain crowd counting. To be specific, firstly, train a source model with traditional supervised learning on synthetic data. Then, exploit a few labeled target data to learn two parameters (factor and bias) for each source neuron. Finally, generate target model neurons from source neurons with a linear transformation. The entire process is shown in Fig. 2.

In summary, the main contributions of this paper are:

- Propose a novel Neuron Linear Transformation method to model the domain shift. It is the first time that the domain shift can be measured at the parameter level.
- Design a newly few-shot learning framework to optimize the domain shift parameters, while few-shot learning in other CDCC methods is exploited to fine-tune the partial layers.
- Achieve more practical results on adapting synthetic dataset to six real-word crowd counting datasets. Further experiments show that NLT can also promote supervised learning performance.

II. Related Work

In this section, we briefly review the relevant works from the three tasks: supervised crowd counting, cross-domain crowd counting, and few-shot learning.

Supervised Crowd Counting. Most of the supervised crowd counting algorithms focus on addressing scale variability in recent years. From the multi-column scale-aware architecture, zhang *et al.* [25] propose a three-columns network with different kernels for scale perception. López-Sastre *et al.* [26] introduce a HydraCN with three-columns, where each column is fed by a patch from the same image with a different scale. Wu *et al.* [27] develop a powerful multi-column scale-aware CNN with an adaptation module to fuse the sparse and congested column. To generate a high-quality density map, AFP [28] fuses the attention map and intermediary density map in each column. ic-CNN [29] delivers the feature and predicted density map from the low-resolution CNN to the high-resolution CNN. Hossain *et al.* [30] propose a multi-column scale-aware attention network, where each column is weighted with the output of a global scale attention network and local scale attention network. Besides, the single-column scale-aware CNN [31], [32] also make great progresses in recent research. CSRNet[33], CAN [34], and FPNCC [35] employ multiple paths only in partly layers, which is a combination of multi-column and single-column scale-aware CNN.

From other perspectives, HA-CNN [36] employs a spatial attention module (SAM) and a set of global attention modules (GAM) to enhance the features extracting ability selectively. To further refine the density map, CRNet [37] stacks several fully convolutional networks together recursively with the previous output as the next input. Every stage utilizes previous density output to refine the predicted density maps gradually. To count people in various density crowds, PaDNet [38] develops three components: Density-Aware Network (DAN) module extracts pan-density information, Feature Enhancement Layer (FEL) effectively captures the global and local contextual features, and Feature Fusion Network (FFN) embeds spatial context and fuses these density-specific features. CLPNet [39] exploits a cross-level parallel network to conduct multi-scale fusion from five different aggregation modules. It extracts multiple low-level features from VGG-16 and then fuses them with specific scale aggregation modules in the high-level stage. To tackle the crowd distribution and background interference problems, Mo *et al.* [40] utilize local information of distance between human heads and the global information of the people distribution in the whole image to achieve head size estimation. The predicted head masks are used to reduce the impact of different crowd scale and background noise.

Cross-domain Crowd Counting. In addition to the exploration mentioned above, CCDC, a new research hotspot, is beginning to interest researchers. This task aims to transfer what the model learns from one dataset to another unseen dataset. Literature [19] is the earliest research in this filed, which establishes a large-scale synthetic dataset to pre-train a model that improves the robust over real-world datasets by fine-tuning. Except fine-tuning, it also trains a counter without using any real-world labeled data. It narrows the domain gap by using the Cycle GAN [41] and SE Cycle GAN [19], [42] to generate a realistic image. Recently, several efforts have been made to follow it, DACC [20], a method for domain adaptation based on image translation and Gaussian-prior reconstruction, achieves new state-of-the-art results on several mainstream datasets. At the same time, some works [22], [21]

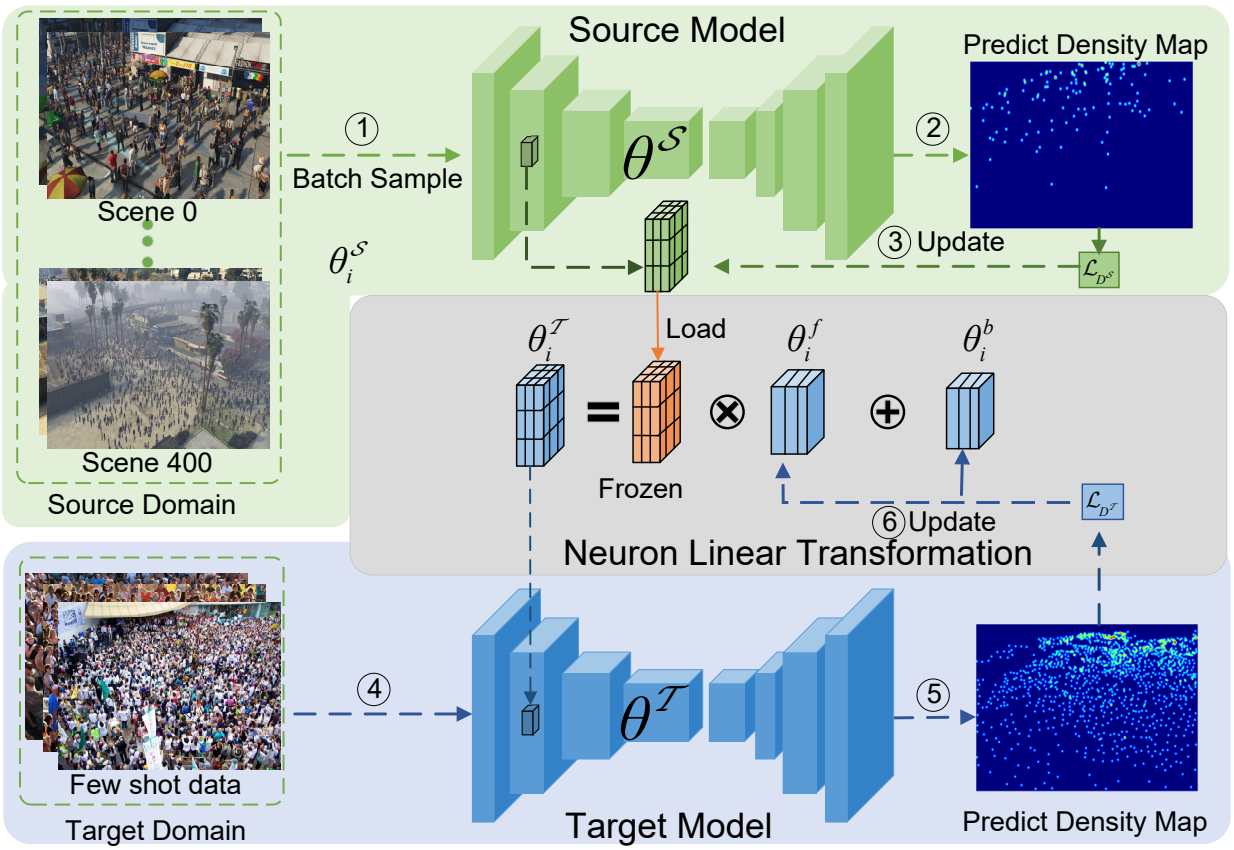


Fig. 2: The flowchart of our proposed NLT, which consists of three components: 1) Source model is trained with the synthetic data; 2) The learnable parameters θ^f and θ^b are used to model the domain shift, which are defined according the source model. Namely, for a source neuron $\theta_i^{\mathcal{S}} \in \theta^{\mathcal{S}}$ (i is the index of neurons), there are a θ_i^f and a θ_i^b that are used to generate a target neuron $\theta_i^{\mathcal{T}}$ by a linear operation. 3) After loading the transferred parameters $\theta^{\mathcal{T}}$ to the target model, the few-shot data are feed into the target model to update the domain shift parameters.

extract domain invariant features based on adversarial learning. Experimental results show that those methods can narrow the domain shift to some extent.

For the supervised crowd counting, the advantage is that it can get well-performed models by conventional training. However, the supervised methods require much manual annotation samples, which are laborious to get them. Besides, the model trained in a specific scenario does not work well when applying to other scenarios because of the domain gap. CDCC can alleviate the above shortcomings, but the current CDCC methods that adapt to real-world scenes from synthetic data cannot achieve similar performance with supervised learning. Overall, the intersection of synthetic data and real-world data proves to be particularly fertile ground for groundbreaking new ideas, and this field will become more significant over time.

Few-shot Learning. Since it involves a small number of target domain samples in our CDCC method, we hereby introduce some studies related to few-shot learning. The few-shot learning is based on prior experience with very similar tasks where we have access to large-scale training sets and then train a deep learning model using only a few training examples. Early few-shot learning methods [43], [44], [45] are based on hand-crafted features. Vinyals *et al.* [46] use a memory component in a neural net to learn common representation from very little data. Snell *et al.* [47] propose Prototypical Networks, which map examples to a dimensional vector space. Ravi and Larochelle [48] use an LSTM-based meta-learner to learn an update rule for training a neural network learner. Model-Agnostic Meta-Learning (MAML) [49] learns a model parameter initialization that generalizes better to similar tasks. Similar to MAML, Mishra *et al.* [50] executes stochastic gradient descent for K iterations on a given task, and then gradually moves the initialization weights in the direction of the weights obtained after the K iterations. Santoro *et al.* [51] propose Memory-Augmented Neural Networks (MANNs) to memorize information about previous tasks and leverage that to learn a learner for new tasks. SNAIL [52] is a generic meta-learner architecture to learn a common feature vector for the training images to aggregate information from past experiences. Most of the above few-shot learning methods are based on classification tasks. Besides, few-shot learning has been applied to many computer vision tasks. Siamese-based trackers can be viewed as an application of one-shot learning [53], [54], [55], [56], [57], [58]. For example, [54] get a learner net by off-line training, and then generate a parameter of pupil net online with one sample. [57] propose a new quadruplet deep network that contains four branches

of the same network with shared parameters. It achieves a more powerful representation by examining the potential connections among the training instances. For crowd counting tasks, [59] proposes a one-shot learning approach for learning how to adapt to a target scene using one labeled example. [60] applies the MAML [49] to learn scene adaptive crowd counting with few-shot learning.

III. Approach

In this section, we first define the problem that we want to solve. Then, the NLT, a linear operation at the neuron level, is designed to model the domain shift. Finally, we introduce how to transfer the source model to the target model with NLT operation. Fig. 2 illustrates the entire framework.

A. Problem Setup

In this paper, we strive to tackle the existing problems for domain adaptive crowd counting from the parameter-level with a transformation. The setting assumes access to a source domain (synthetic data) with N_S labeled crowd images $D^{\mathcal{S}} = \{I_i^{\mathcal{S}}, Y_i^{\mathcal{S}}\}_i^{N_S}$. Besides, a target domain (real scene data) provides $N_{\mathcal{S}}$ few-shot images with the labeled density maps $D^{\mathcal{T}} = \{I_i^{\mathcal{T}}, Y_i^{\mathcal{T}}\}_i^{N_{\mathcal{T}}}$. The purpose is to train a source domain model $\mathcal{S}$ with the parameters $\theta^{\mathcal{S}}$ exploiting the $D^{\mathcal{S}}$, and learn a representable domain shift according to $D^{\mathcal{T}}$ with few-shot learning, which are parameterized by the domain factors θ^f and domain biases θ^b. Finally, generating a well performed target model $\mathcal{T}$ with the parameters $\theta^{\mathcal{T}}$ by combining the source model with the domain shift parameters.

B. Neuron Linear Transformation

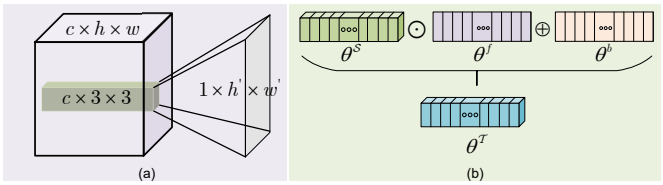


Fig. 3: (a) is the schematic diagram of neuron's definition. (b) shows how to transfer source neuron $\theta^{\mathcal{S}}$ to target neuron $\theta^{\mathcal{T}}$ with NLT.

Inspired by the scale and shift operation [61], we propose a Neuron Linear Transformation method to describe the domain gap, which makes the domain gap visible. To model the domain shift, we assume that *the source model and the target model belong to the same linear space V^n. Each neuron in the target model can be transferred from the corresponding neuron in the source model by a linear transformation.* As shown in Fig. 3 (a), a neuron is defined as a convolutional group used to generate a channel $(1 \times h' \times w')$ in the next layer from the last layer $(c \times h \times w)$. The size of neurons is determined by two factors: the number of channels in the last layer of the CNN and the size of the convolution kernel used to generate the next layer. Fig. 3 (b) shows how the target model's parameters are transferred from the source model by a linear operation. For the source model's neuron $\theta^{\mathcal{S}} \in \mathbb{R}^{c\times h\times w}$, we define the corresponding domain factor $\theta^f \in \mathbb{R}^{c\times 1\times 1}$ and domain bias $\theta^b \in \mathbb{R}^{c\times 1\times 1}$. Then the neuron-level linear transformation can be formulated as follow.

$$\theta^{\mathcal{T}} = \left[f^1 \times \begin{bmatrix} a_{11}^1 & \cdots & a_{1w}^1 \\ \vdots & \ddots & \vdots \\ a_{1h}^1 & \cdots & a_{hw}^1 \end{bmatrix} + \left[b^1\right], \cdots, f^c \times \begin{bmatrix} a_{11}^c & \cdots & a_{1w}^c \\ \vdots & \ddots & \vdots \\ a_{1h}^c & \cdots & a_{hw}^c \end{bmatrix} + b^c \right], \tag{1}$$

where $f^i \in \theta^f, b^i \in \theta^b$ and $a_{hw}^i \in \theta^{\mathcal{S}} \quad (i = 1, 2, \cdots, c)$. The domain adaptation method has two advantages: 1) The target model inherits the good feature extraction ability from the source model and preserves the generalization. 2) Compared with fine-tuning operation, fewer parameters need to be optimized in the target model with NLT. So it reduces the probability of overfitting for few-shot learning in the target domain.

C. Modeling the Domain Shift

Firstly, we introduce a crowd counter to test the NLT. Following the previous work [19], [21], [20], [22]. we take the VGG architecture as the feature extractor. As shown in Fig. 2, the first ten layers of VGG-16 [62] are adopted as the backbone in the encoder stage. That is, the width and height of the output feature are 1/8 of the input image. In the decoder stage, a 3x3 convolutional layer is used to reduce the feature channels to a half. Then an up-sampling layer is followed by a 3x3 convolutional layer to reduce channels. After three repetitions, a 1x1 convolutional layer outputs the prediction density map. The source domain model's training is similar to that of the traditional supervised crowd counting network, except that the training data adopts the synthetic dataset. The $\theta^{\mathcal{S}}$ are optimized by gradient descent as follow,

$$\tilde{\theta}^{\mathcal{S}} = \theta^{\mathcal{S}} - \alpha \nabla \mathcal{L}_{D^{\mathcal{S}}}\left(\theta^{\mathcal{S}}\right), \tag{2}$$

where $\mathcal{L}_{D^{\mathcal{S}}}\left(\theta^{\mathcal{S}}\right) = \frac{1}{2n}\sum_i \left\|\mathcal{S}(I_i^{\mathcal{S}}; \theta^{\mathcal{S}}) - Y_i^{\mathcal{S}}\right\|_2^2$ is a standard MSE loss. n is the batch size of source model. $\mathcal{S}(I_i^{\mathcal{S}}; \theta^{\mathcal{S}})$ is the source model prediction of the i^{th} training data. α denotes the learning rate.

Secondly, we introduce how to embed NLT into our target model training. As shown in Fig. 2, the target model keeps the same architecture as the source model, but the parameters in the target model are transferred from the source model. So, the number of learnable parameters is different. Taking the VGG-16 [62] backbone as an example, the channels of the first ten layers are $\{64, 64, 128, 128, 256, 256, 256, 512, 512, 512\}$. According to the neuron's definition in III-B, the total number of neurons is 2688, which is the sum of the channels' number. Assuming that the source model contains k neurons, each neuron needs a factor and a bias in the target model. So the number of the learnable parameters θ^f and θ^b is $2 \times k$. As the convolution kernel of VGG-16 is 3×3, learnable parameters in the target model are $\sim 2/9$ of the source model. The θ^f and θ^b are defined to model domain shift by neuron-level linear

transformation. Specifically, we model the domain shift by transferring all neurons in the source model to the target model with the proposed NLT. According to Equ. (1), the mapping can be expressed as follows,

$$\theta_i^{\mathcal{T}} = \theta_i^{\mathcal{S}} \odot \theta_i^f \oplus \theta_i^b, (i = 1, 2, \cdots, k), \tag{3}$$

where θ_i^f represents the domain shift factor, initialized by 1. θ_i^b represents the domain shift bias, which is initialized to 0.

Since we introduce the learnable parameters to describe the task gap, some target domain labeled images are needed to optimize the parameters. However, within the requirement of domain adaptation, we only use a few data to support the training. During training the source model, $\theta^{\mathcal{S}}$ are learned. However, they will be frozen when the target model is updated. After the calculation of Equ. 3, $\theta^{\mathcal{T}}$ participate in the feed-forward of the target model. Therefore, only the gradients of θ^f and θ^b need to be calculated in the back-forward process. That is, θ^f and θ^b are learned in the target model. The loss for optimizing the parameters is defined as follows,

$$\begin{aligned}\mathcal{L}_{D^{\mathcal{T}}}\left(\theta^{\mathcal{T}}\right) =& \frac{1}{2n}\sum_i \left\|\mathcal{T}(I_i^{\mathcal{T}};\theta^{\mathcal{T}}) - Y_i^{\mathcal{T}}\right\|_2^2 + \\ & \lambda(\sum_{i=1}^{k}(\theta_i^f - 1)^2 + (\theta_i^b)^2),\end{aligned} \tag{4}$$

where the former term is the density estimated loss of the few-shot data. It is the same as the loss of the source model. $(I_i^{\mathcal{T}}, Y_i^{\mathcal{T}}) \in D^{\mathcal{T}}$ is the i^{th} input image and density map. $\mathcal{T}(I_i^{\mathcal{T}};\theta^{\mathcal{T}})$ is the prediction density map. The latter term is the L2 regularization loss of parameters θ^f and θ^b, with the purpose of preventing overfitting $D^{\mathcal{T}}$ in the target domain. λ is the weighted parameter. Finally, the target model is optimized as follows,

$$\tilde{\theta}^{\mathcal{T}} = \theta^{\mathcal{T}} - \beta\nabla\mathcal{L}_{D^{\mathcal{T}}}\left(\theta^{\mathcal{T}}\right), \tag{5}$$

where β denotes the learning rate of target model.

IV. Implementation Details

Executive Stream. In the training phase, the workflow is shown in Fig. 2 ① ∼ ⑥, once iteration updates parameters for two models. Firstly, as shown in ① ∼ ③, $\theta^{\mathcal{S}}$ are updated according to a batch sampling from the GCC dataset. Secondly, in ④ ∼ ⑥, the domain shift parameters are updated with the few-shot data provided in the target domain. Finally, the parameters of the target model are obtained by NLT. In the testing phase, we use the best-performing model on the validation set to make an inference.

Parameter Setting. In each iteration, we input 12 synthetic images and 4 target few-shot images. Adam algorithm [63] is performed to optimize the networks. The learning rate α in the source model and β in the target model are initialized as 10^{-5}. The parameter λ for target model loss function in Eq. 4 is fixed to 10^{-4}. Our code is developed based on the C^3 Framework [64].

Scene Regularization. In some domain adaptation fields, such as semantic segmentation, the label distribution is highly consistent in two domains. Unlike that, current real-world crowd datasets are very different in density range, such as the number of the people in MALL [24] dataset is ranging from 13 to 53, but the GCC [19] dataset is ranging from 0 to 3,995. For avoiding negative adaptation by the different density ranges, we adopt a scene regularization strategy proposed by [19] and [22]. In other words, we add some filter conditions to select proper synthetic images from GCC as the source domain data for different real-world datasets.

V. Experiments

In this section, we first introduce the evaluation metrics and the selected datasets, and then a comprehensive ablation study is performed to verify the effectiveness of our proposed method. Next, we analyze the shifting phenomenon in adopting synthetic dataset to different real-world datasets from statistics. Besides, we also discuss the effect of selected few-shot data on performance improvement. Finally, we present the testing results and visualization results of our method in six real-world datasets.

A. Evaluation Criteria

Counting Error. According to the evaluation criteria widely used in crowd counting, the counting error is usually reflected in two metrics, namely Mean Absolute Error(MAE) and Mean Square Error(MSE). MAE measures the mean length of the predicted Error, while MSE measures the model's robustness to outliers. They are defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|^2}, \tag{6}$$

where N is the number of images to be tested, and y_i and $\hat{y}_i$ are the ground truth and estimated number of people corresponding to the i^{th} sample, which is obtained by summing all the pixel values in the density map.

Density Map Quality. To further evaluate the quality of density maps, we also calculate PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity in Image) [65]. For those two metrics, the larger the value, the higher the quality of the predicted density maps.

B. Datasets

The synthetic dataset GCC [19] is the only source domain in this paper. As for the target domain, to ensure the sufficiency of our experiments, we respectively select two datasets from high-level density, medium-level density and low-level density datasets, a total of six datasets, namely UCF-QNRF [13], Shanghai Tech Part A [25], Shanghai Tech Part B [25], WorldExpo'10 [66] , Mall [23] and UCSD [24].

Source Domain Dataset. *GCC* is a large-scale synthetic dataset, which is sampled from 400 virtual scenes by a computer mod. It contains 15,212 of accurately annotated images with a total of 7,625,843 instances. There is an average of 501 people in each image.

Congested Crowd Dataset. *UCF-QNRF* is collected from a shared image website. Therefore, the dataset contains a variety of scenes. It consists of 1,535 images(1201 training

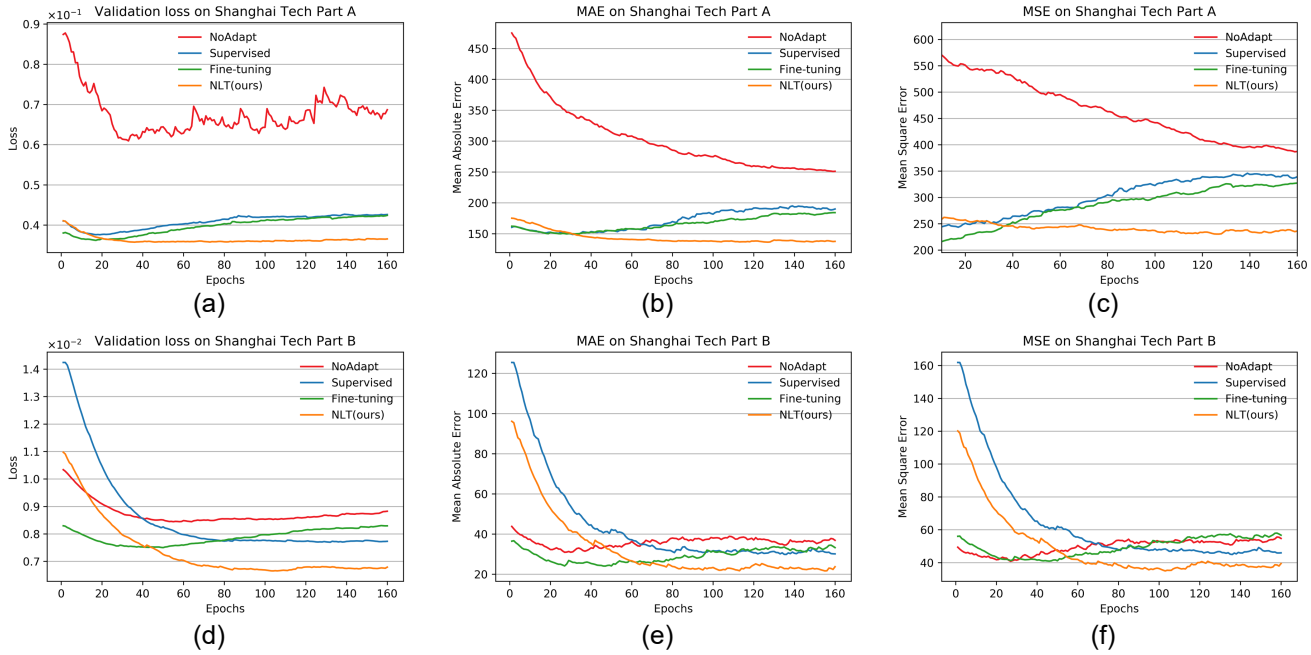


Fig. 4: The effects of our NLT and other training methods on learning process and performance. (a)(b)(c) and (d)(e)(f) show the validation loss and performance on Shanghai Tech Part A and B dataset, respectively.

TABLE I: The performance of different training methods on Shanghai Tech Part A and Shanghai Tech Part B.

Method	DA	FS	Shanghai Tech Part A				Shanghai Tech Part B			
			MAE	MSE	PSNR	SSIM	MAE	MSE	PSNR	SSIM
NoAdpt	✗	✗	188.0	279.6	20.91	0.670	20.1	29.2	26.62	0.895
Supervised	✗	✔	107.2	165.9	21.53	0.623	16.0	26.7	26.8	0.932
Fine-tuning a	✔	✔	105.7	167.6	21.72	0.702	13.8	22.3	27.0	0.931
Fine-tuning b	✔	✔	110.8	167.4	20.98	0.722	13.1	20.8	27.14	0.924
NLT (ours)	✔	✔	93.8	157.2	21.89	0.729	11.8	19.2	27.58	**0.937**
IFS [20]+NLT (ours)	✔	✔	**90.1**	**151.6**	**22.01**	**0.741**	**10.8**	**18.3**	**27.69**	0.932

and 334 testing images), with $1,251,642$ annotated instances. The average number of people is 815 per image. *Shanghai Tech Part A* is also randomly collected from the Internet with different scenarios. It consists of 482 images (300 training and 182 testing images) with different resolutions. The average number of people in an image is 501.

Moderate Crowd Dataset. *Shanghai Tech Part B* is captured from the surveillance camera on the Nanjing Road in Shanghai, China. It contains 716 samples (400 training and 316 testing images). The scenes are relatively consistent, with an average of 123 people per picture. *WorldExpo'10* consists of $3,980$ labeled images. They are collected from 108 surveillance scenes (103 scenes for training and the remaining 5 scenes for testing) in Shanghai 2010 WorldExpo. The average number of people is 50 per image.

Sparse Crowd Dataset. *Mall* is captured from a surveillance camera installed in a shopping mall, which records the $2,000$ (800 for training and $1,200$ for testing) sequential frames. The average of people in each image is 31. *UCSD* consists of $2,000$ frames (frames $601-1,400$ for training and the others for testing), which are collected from a single-scene surveillance video. The average number of pedestrian in each image is 25.

C. Ablation Study

We present our ablation experiments from two perspectives. Firstly, regarding the few-shot data, we demonstrate the impact by using different training methods. Secondly, for the proposed NLT, we discuss the effects of θ^f and θ^b on modeling the domain shift. The following experiments are conducted on Shanghai Tech Part A and B datasets, and the selected few-shot data are both the 10% of the training set.

Compared with Other Training Methods. Six training methods are used to demonstrate the role of few-shot data in narrowing the domain gap. The specific settings are as follows:

- **NoAdpt.** Train the model on the GCC dataset.
- **Supervised.** Train the model on few-shot data.
- **Fine-tuning a.** Train the model on the GCC dataset and fine-tune all parameters with few-shot data.
- **Fine-tuning b.** Train the model on the GCC dataset and fine-tune the decoder (last four layers) with few-shot data.
- **NLT (ours).** Train the model from GCC to the real-world dataset with our NLT and training strategy.

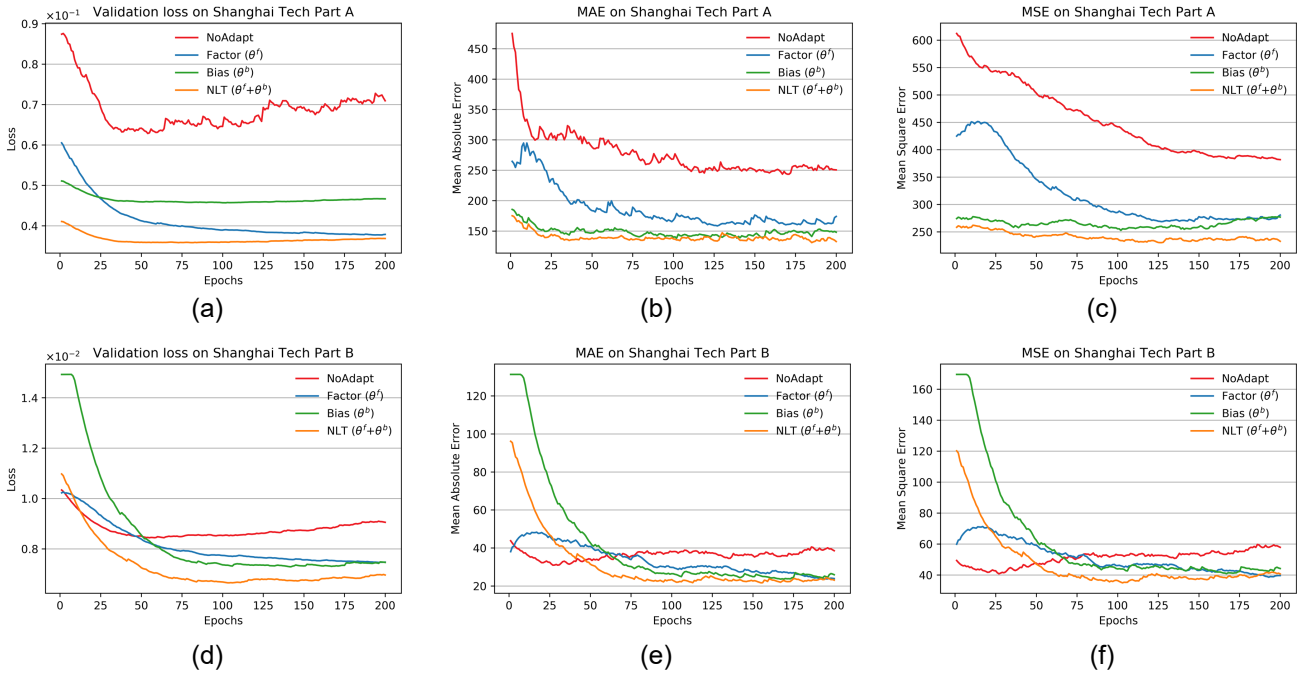


Fig. 5: The effects of the domain shift parameters θ^f and θ^b. (a)(b)(c) and (d)(e)(f) show the validation loss and performance on Shanghai Tech Part A and B dataset, respectively.

TABLE II: The effectiveness of the domain shift parameters θ^f and θ^b on the testing set of Shanghai Tech Part A and B.

Method	DA	FS	Shanghai Tech Part A				Shanghai Tech Part B			
			MAE	MSE	PSNR	SSIM	MAE	MSE	PSNR	SSIM
NoAdpt	✗	✗	188.0	279.6	20.91	0.670	20.1	29.2	26.62	0.895
Fine-tuning	✔	✔	105.7	167.6	21.72	0.702	13.8	22.3	27.0	0.931
Factor (θ^f)	✔	✔	109.2	161.3	21.49	0.758	13.5	23.5	27.26	0.921
bias (θ^b)	✔	✔	107.8	169.9	21.14	0.796	12.8	20.6	27.17	0.916
NLT ($\theta^f + \theta^b$)	✔	✔	**93.8**	**157.2**	**21.89**	**0.729**	**11.8**	**19.2**	**27.58**	**0.937**

- **IFS+NLT (ours).** Replace the original GCC data with IFS [20] translated GCC [19] in the last setting.

As shown in Fig. 4, we draw the loss and performance curves on the validation set during training. Taking Shanghai Tech Part A dataset as an example, it is difficult to reduce the loss of the validation set without domain adaptation. The supervised training and fine-tuning with few-shot data can significantly reduce the loss, but it is easy to suffer from overfitting. Compared to supervised training and fine-tuning, NLT can reach lower validation loss and inhibit overfitting. In Fig. 4 (b) and (c), the MAE and MSE curves also illustrate the effectiveness of NLT. Similarly, in Fig. 4 (d), (e) and (f), Shanghai Part B have the same trend, which proves that NLT is effective for both dense and sparse scenes.

Table I shows the testing results of six training methods. No adaptation is usually hard to achieve satisfying results, which validates the vast distance between real and synthetic data. As shown in lines 3, 4, and 5, fully supervised training and fine-tuning on a pre-trained GCC model with few-shot data yield better results than no adaptation. It shows that few-shot data has a significant effect on narrowing the domain gap. By comparing the results of lines 4, 5, and 6, the proposed NLT yields better performance than the fine-tuning operation. Taking MAE as an example, in Shanghai Tech part A, NLT reduces the counting error by 13.1% compared with fine-tuning a. In Shanghai Tech part B, it reduces by 10.0%. We also test NLT on the stylized images generated by IFS [20], and the results show that NLT can help other domain adaptation methods to improve performance further. In conclusion, the proposed NLT maximizes the potential of few-shot data.

The Influence of Domain Shift Parameters. For the proposed NLT, two types of parameters are defined to learn the transformation of neurons, namely θ^f and θ^b. To verify the parameters' validity and compatibility, we use factor θ^f, bias θ^b, and both of them to model the model shift, respectively. The details of the experiments are shown in Fig .5.

The red curves represent no domain adaptation results. The loss curve is descending at the beginning of the training. However, as time goes on, it changes from decrease to increase. The reason is that the model trained with synthetic data has a limited ability to fit the real data. Once the limit value is passed, the model will continuously deviate from the target domain. The blue and green curves show the effectiveness

TABLE III: The performance of other domain adaptation (DA) methods and the proposed NLT on the six real-world datasets. FS refers to 10% shot data from the target domain.

Method	Backbone	DA	FS	Shanghai Tech Part A				Shanghai Tech Part B				UCF-QNRF			
				MAE	MSE	PSNR	SSIM	MAE	MSE	PSNR	SSIM	MAE	MSE	PSNR	SSIM
CycleGAN [41]	VGG-16	✔	✗	143.3	204.3	19.27	0.379	25.4	39.7	24.60	0.763	257.3	400.6	20.80	0.480
SE CycleGAN [19]	VGG-16	✔	✗	123.4	193.4	18.61	0.407	19.9	28.3	24.78	0.765	230.4	384.5	21.03	0.660
FA [22]	VGG-16	✔	✗	-	-	-	-	16.0	24.7	-	-	-	-	-	-
FSC [21]	VGG-16	✔	✗	129.3	187.6	21.58	0.513	16.9	24.7	26.20	0.818	221.2	390.2	**23.10**	0.7084
IFS [20]	VGG-16	✔	✗	112.4	176.9	21.94	0.502	13.1	19.4	**28.03**	0.888	211.7	357.9	21.94	0.687
NoAdpt (ours)	VGG-16	✗	✗	188.0	279.6	20.91	0.670	20.1	29.2	26.62	0.895	276.8	453.7	22.22	0.692
NLT (ours)	VGG-16	✔	✔	93.8	157.2	21.89	0.729	11.8	19.2	27.58	0.937	172.3	307.1	22.8	0.729
IFS[20]+NLT (ours)	VGG-16	✔	✔	**90.1**	**151.6**	**22.01**	0.741	10.8	**18.3**	27.69	0.932	**157.2**	**263.1**	23.01	**0.744**
NLT (ours)	ResNet-50	✔	✔	91.4	153.4	21.45	**0.749**	**10.4**	18.8	27.79	**0.942**	165.8	279.7	22.89	0.734

Method	Backbone	DA	FS	WorldExpo'10 (only MAE)						UCSD				MALL			
				S1	S2	S3	S4	S5	Avg.	MAE	MSE	PSNR	SSIM	MAE	MSE	PSNR	SSIM
CycleGAN [41]	VGG-16	✔	✗	4.4	69.6	49.9	29.2	9.0	32.4	-	-	-	-	-	-	-	-
SE CycleGAN [19]	VGG-16	✔	✗	4.3	59.1	43.7	17.0	7.6	26.3	-	-	-	-	-	-	-	-
FA [22]	VGG-16	✔	✗	5.7	59.9	19.7	**14.5**	8.1	21.6	2.0	2.43	-	-	2.47	3.25	-	-
IFS [20]	VGG-16	✔	✗	4.5	33.6	**14.1**	30.4	4.4	17.4	1.76	2.09	24.42	0.950	2.31	2.96	25.54	0.933
NoAdpt (ours)	VGG-16	✗	✗	5.0	89.9	63.1	20.8	17.1	39.2	12.79	13.22	23.94	0.899	6.20	6.96	24.65	0.879
NLT (ours)	VGG-16	✔	✔	2.3	22.8	16.7	19.7	3.9	13.1	1.58	1.97	25.29	0.942	1.96	2.55	26.92	**0.967**
IFS[20]+NLT (ours)	VGG-16	✔	✔	**2.0**	**15.3**	14.7	18.8	3.4	**10.8**	1.48	1.81	**25.58**	**0.965**	1.86	**2.39**	**27.03**	0.944
NLT (ours)	ResNet-50	✔	✔	3.1	17.8	17.9	20.6	**3.2**	12.5	**1.42**	**1.76**	25.56	0.964	**1.80**	2.42	26.84	0.940

of domain factor θ^f and domain bias θ^b, respectively, both of them can significantly reduce losses and improve performance. It is worth noting that factor is not easy to overfit, but the convergence is slow, while bias converges faster but is easy to overfit. When the two are together, they complement each other and perform best.

The results of the test set are shown in Table II. The learnable parameters for factors θ^f and bias θ^b both are $\sim 1/9$ of the source model. Fine-tuning updates all parameters of the source model. For example, in Shanghai Tech Part A, 10% of the training set are treated as few-shot data, factor θ^f and bias θ^b achieve the similar results compared with fine-tuning, which verify that it is effective to use factor and bias to represent domain shift. We achieve the best results when combining them as NLT.

D. Adaptation Results on Real-world Datasets

In this section, we test the performance of the NLT by using it to learn the domain shift from GCC to six real-world datasets and compare it with the other domain adaptation methods.

Metrics Report. Table III lists the statistical results of the four metrics (MAE↓/MSE↓/PSNR↑/SSIM↑). Compared with the image translation (CycleGAN [41], SE CycleGAN [19], and IFS [20]) and feature adversarial learning (FA [22] and FSC [21]) methods, our method performs better with the use of 10% annotated data in the target domain. Taking MAE as an example, as the lavender row shows, NLT reduce counting errors by 16.5%, 10.0%, 18.6%, 29.9%, 10.0%, and 15.2% compared with the above methods on six real-world datasets, respectively. As for the quality of predicted density map, we also achieve a significant improvement on PSNR and SSIM, indicating that few-shot data make a great contribution to noise cancellation in the background region. Experiments with different density datasets demonstrate the universality of NLT for cross-domain counting tasks.

Furthermore, we discuss the combination of NLT and other domain adaptation methods. We implement stylistic realism for the GCC [19] dataset by using IFS [20], which is a image translation method for cross-domain crowd counting. These images are then treated as source domain data, and the proposed NLT is applied to achieve the domain adaptation. The final test results in the six real-world datasets are shown in Table III, lavender row. Compared with the original IFS [20], the NLT decreases the MAE by 19.8%, 17.6%, 25.7%, 37.9%, 16.0%, and 19.5% on the six real data sets, respectively. We also test NLT on the ResNet50 architecture, and the test results are shown in Table III, light cyan row. On the whole, the ResNet50 backbone outperforms the VGG-16 backbone because of its deeper structure and superior semantic extraction capability.

Visualization Results. Fig. 6 shows the visualization results of no adaptation and the proposed NLT. Column 3 shows the results without domain adaptation. The regression results are not acceptable in a congested scene like Shanghai Tech Part A, especially the gray-scale image in Row 2. On Shanghai Tech Part B, the counting results of no domain adaptation are a little close to the ground truth, but the problems remain in detail and background. The red box in Row 3 shows that the regression value is weak in some areas. The estimation errors in the background region also prevent the performance, such as the red box shown on Raw 4. After the domain adaptation, the above questions are alleviated. In general, the NLT improves the density map in counting values and details. To more intuitively demonstrate our domain adaptation effect, we show more results in Fig. 9. From the performance on different datasets, NLT is effective for cross-domain counting tasks with different crowding levels.

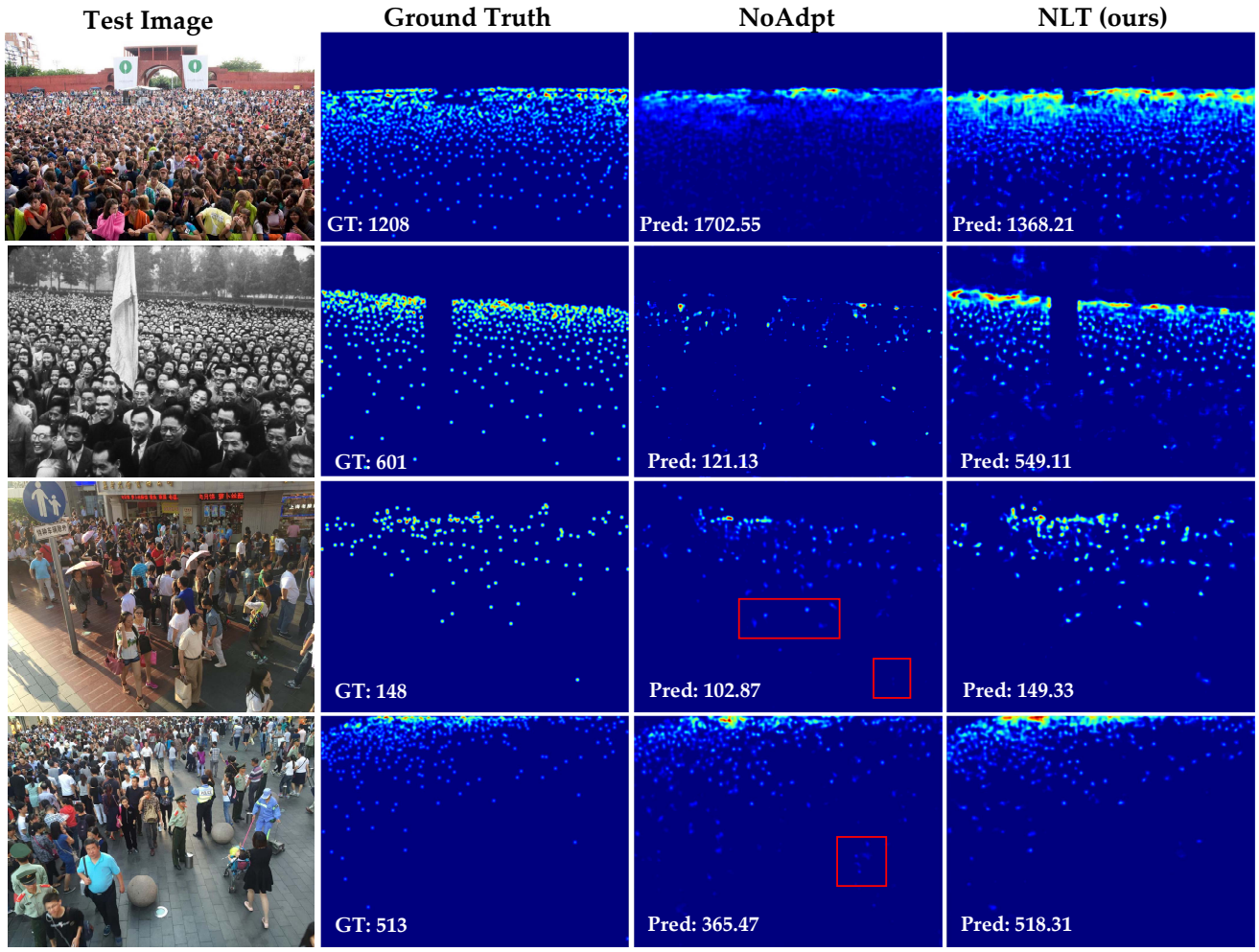


Fig. 6: Exemplar results of adaptation from GCC to Shanghai Tech Part A/B dataset. Row 1 and 2 come from Shanghai Tech Part A, and others are from Part B.

Computational Statistics. Table IV shows the computation statistics on the proposed NLT and several other DA methods, including parameter size, Giga Floating-point Operations Per second (GFLOPs), and Frames Per Second (FPS). All the methods are tested by inputting images with a size of 768x1024 under a single GTX-1080Ti GPU. Since SE CycleGAN [19] and IFS [20] involve image transformation in the training phase, we only compute the crowd counters in all methods here. All crowd counters use VGG-16 as the backbone. NLT ranks below the IFS because the NLT introduces the domain shift parameters θ^f and θ^b. The advantage of NLT is that the training process is faster as IFS needs image transfer.

TABLE IV: The computational statistics of the proposed NLT and other domain adaptation methods.

Method	SE CycleGAN[19] FA[22]	FSC[21]	IFS[20]	NLT(ours)
params(MB)	67.1	68.6	**30.1**	38.1
GFLOPs	326.8	311.8	**236.9**	278.9
FPS	9.3	9.6	**18.3**	16.7

E. Statistical Analysis of Domain Shift

Domain factor θ^f and domain bias θ^b are defined as the parameters to model the shift from the source domain to the target domain. They are initialized to 1 and 0, respectively, and optimized to narrow the domain gap by few-shot data. In Sec. V-C, we verify its effectiveness by testing it on different datasets. In this section, we will further analyze the significance of these parameters from mathematical statistics.

There are 15 convolutional layers in the VGG-16 backbone and the decoder, and each convolution kernel contains a domain factor and domain bias parameter. For the well-trained target model, we calculate the mean values of factor and bias at each layer. The statistical results are shown in Fig. 7, where the mean value for factor is subtracted from the initial value 1. As Fig. 7 (a) shown, at most layers, the mean value of factor and bias are less than 1 and 0, respectively. Therefore, the effect of factor θ^f and bias θ^b is to shrink the parameters of the source model. We call this shift a "down domain shift". The distribution of UCF-QNRF in Fig. 7 (b) is similar to that of Shanghai Tech Part A. Both of them are collected from the internet, so it has a similar distribution. In Fig. 7 (c), the averages of factor and bias are greater than 1 and 0 in most

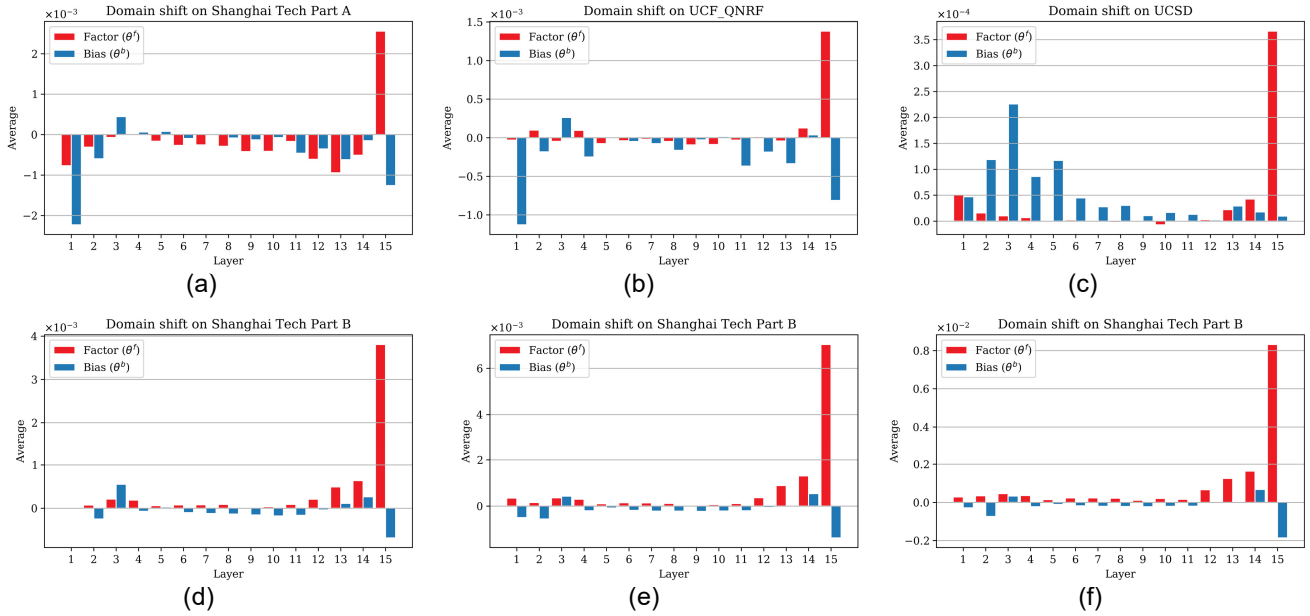


Fig. 7: The averages of domain factor and domain bias in each layer of the network.

layers, respectively. We define the shift as "up domain shift." In Fig. 7 (d), factor and bias are distributed on both sides. We define this shift as "up-down domain shift". In addition, in Fig. 7 (d)(e)(f), we use 10%, 30%, and 50% of the training set as few-shot data to learn domain shift parameters. The distributions are the same eventually. This reveals that only a few target domain labeled images are needed to learn the representation of domain shift.

F. Analysis in Selecting Few-shot Data

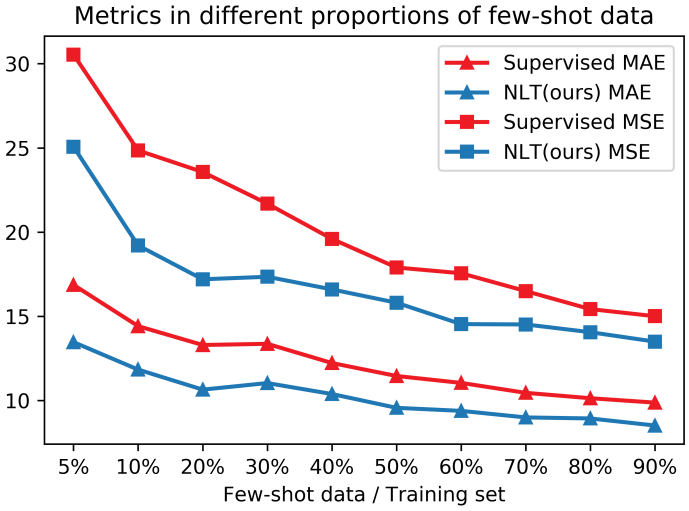


Fig. 8: The testing results for NLT (blue) and supervised training (red) with different ratios of few-shot data on Shanghai Tech Part B. The triangle and rectangle represent MAE and MSE, respectively.

Since our domain adaptation method requires a few target domain labeled images, in this section, we will discuss the effects of selecting different proportions of few-shot data for NLT. As shown in Fig. 8, we conduct the experiments on the Shanghai Tech Part B dataset. The horizontal axis represents different proportions of the few-shot images. The vertical axis represents the metric values on the Shanghai Tech Part B testing set. The experiments illustrate that NLT performs better with increasing few-shot learning data. Besides, compared with the traditional supervised training methods, the proposed NLT is better in every configuration. It shows that NLT can improve the generalization ability of the supervised model. Overall, the NLT is robust in the different shot data settings.

VI. Conclusions

This paper summarizes the existing problems of CDCC methods and rethink the domain shift from model-level. To convert the source model to the target model, we propose a Neuron Linear Transformation (NLT) method to model the domain shift and optimize the domain shift parameters by few-shot learning. Extensive experiments show that the NLT achieves comparable performance with other domain adaptation methods by using 10% target domain data. Besides, it also has a better expression ability for domain shift. The proposed NLT also be applied in other domain adaptation tasks in future work, such as semantic segmentation, pedestrian Re-ID, and saliency object detection. Take the saliency object detection task [67], [68], [69] as an example, it is similar to binary segmentation task, which needs either bounding boxes or pixel-level annotations to supervise. We can also utilize computer mod to synthesize numerous images and annotate the saliency region automatically. The NLT proposed in this paper can be employed to model the domain gap of cross-domain saliency object detection.

References

[1] H. Idrees, K. Soomro, and M. Shah, "Detecting humans in dense crowds using locally-consistent scale prior and global occlusion reasoning,"

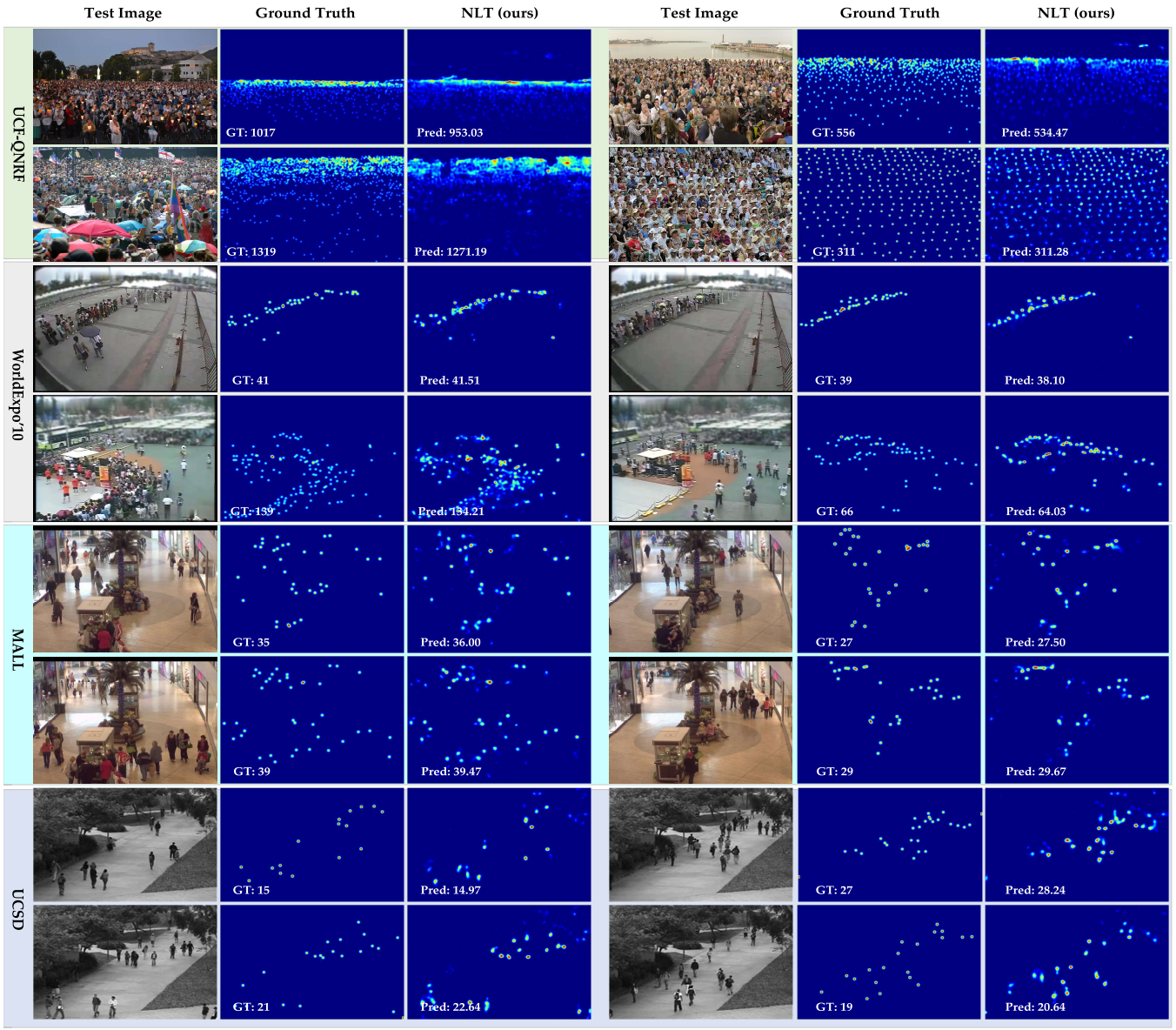


Fig. 9: More visual samples of adaptation from GCC to other four real-world datasets with our proposed NLT.

IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 10, pp. 1986–1998, 2015.

[2] Q. Wang, M. Chen, F. Nie, and X. Li, "Detecting coherent groups in crowd scenes by multiview clustering," *T-PAMI*, vol. 42, no. 1, pp. 46–58, 2020.

[3] E. Heim, A. Seitel, J. Andrulis, F. Isensee, C. Stock, T. Ross, and L. Maier-Hein, "Clickstream analysis for crowd-based object segmentation with confidence," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2814–2826, 2017.

[4] S. Gao, Q. Ye, J. Xing, A. Kuijper, Z. Han, J. Jiao, and X. Ji, "Beyond group: multiple person tracking via minimal topology-energy-variation," *IEEE Transactions on Image Processing*, vol. 26, no. 12, pp. 5575–5589, 2017.

[5] S. Huang, X. Li, Z. Zhang, F. Wu, S. Gao, R. Ji, and J. Han, "Body structure aware deep crowd counting," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1049–1059, 2017.

[6] V. A. Sindagi and V. M. Patel, "Ha-ccn: Hierarchical attention-based crowd counting network," *IEEE Transactions on Image Processing*, vol. 29, pp. 323–335, 2019.

[7] M. Ling and X. Geng, "Indoor crowd counting by mixture of gaussians label distribution learning," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5691–5701, 2019.

[8] Y. Tian, Y. Lei, J. Zhang, and J. Z. Wang, "Padnet: Pan-density crowd counting," *IEEE Transactions on Image Processing*, 2019.

[9] X. Jiang, L. Zhang, P. Lv, Y. Guo, R. Zhu, Y. Li, Y. Pang, X. Li, B. Zhou, and M. Xu, "Learning multi-level density maps for crowd counting," *IEEE transactions on neural networks and learning systems*, 2019.

[10] B. Zhao, X. Li, and X. Lu, "Property-constrained dual learning for video summarization," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[11] B. Zhao, X. Li, and X. Lu, "Cam-rnn: Co-attention model based rnn for video captioning," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5552–5565, 2019.

[12] S. A. M. Saleh, S. A. Suandi, and H. Ibrahim, "Recent survey on crowd density estimation and counting for visual surveillance," *Engineering Applications of Artificial Intelligence*, vol. 41, pp. 103–114, 2015.

[13] 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.

[14] 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.

[15] M. Zhao, J. Zhang, C. Zhang, and W. Zhang, "Leveraging heterogeneous auxiliary tasks to assist crowd counting," in *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, 2019, pp. 12 736–12 745.

[16] J. Wan, W. Luo, B. Wu, A. B. Chan, and W. Liu, "Residual regression with semantic prior for crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4031–4040.

[17] J. Wan and A. Chan, "Adaptive density map generation for crowd counting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1130–1139.

[18] 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.

[19] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8198–8207.

[20] J. Gao, T. Han, Q. Wang, and Y. Yuan, "Domain-adaptive crowd counting via inter-domain features segregation and gaussian-prior reconstruction," *arXiv preprint arXiv:1912.03677*, 2019.

[21] T. Han, J. Gao, Y. Yuan, and Q. Wang, "Focus on semantic consistency for cross-domain crowd understanding," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 1848–1852.

[22] J. Gao, Y. Yuan, and Q. Wang, "Feature-aware adaptation and density alignment for crowd counting in video surveillance," *IEEE Transactions on Cybernetics*, pp. 1–12, 2020.

[23] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting." in *BMVC*, vol. 1, no. 2, 2012, p. 3.

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

[25] 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.

[26] D. Onoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 615–629.

[27] X. Wu, Y. Zheng, H. Ye, W. Hu, J. Yang, and L. He, "Adaptive scenario discovery for crowd counting," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 2382–2386.

[28] D. Kang and A. Chan, "Crowd counting by adaptively fusing predictions from an image pyramid," *arXiv preprint arXiv:1805.06115*, 2018.

[29] V. Ranjan, H. Le, and M. Hoai, "Iterative crowd counting," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 270–285.

[30] M. Hossain, M. Hosseinzadeh, O. Chanda, and Y. Wang, "Crowd counting using scale-aware attention networks," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2019, pp. 1280–1288.

[31] 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.

[32] L. Zhang, M. Shi, and Q. Chen, "Crowd counting via scale-adaptive convolutional neural network," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2018, pp. 1113–1121.

[33] 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.

[34] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5099–5108.

[35] T. W. Cenggoro, A. H. Aslamiah, and A. Yunanto, "Feature pyramid networks for crowd counting," *Procedia Computer Science*, vol. 157, pp. 175–182, 2019.

[36] V. A. Sindagi and V. M. Patel, "Ha-ccn: Hierarchical attention-based crowd counting network," *IEEE Transactions on Image Processing*, vol. 29, pp. 323–335, 2020.

[37] Y. Liu, Q. Wen, H. Chen, W. Liu, J. Qin, G. Han, and S. He, "Crowd counting via cross-stage refinement networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 6800–6812, 2020.

[38] Y. Tian, Y. Lei, J. Zhang, and J. Z. Wang, "Padnet: Pan-density crowd counting," *IEEE Transactions on Image Processing*, vol. 29, pp. 2714–2727, 2020.

[39] J. Li, Y. Xue, W. Wang, and G. Ouyang, "Cross-level parallel network for crowd counting," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 566–576, 2020.

[40] H. Mo, W. Ren, Y. Xiong, X. Pan, Z. Zhou, X. Cao, and W. Wu, "Background noise filtering and distribution dividing for crowd counting," *IEEE Transactions on Image Processing*, vol. 29, pp. 8199–8212, 2020.

[41] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint*, 2017.

[42] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Pixel-wise crowd understanding via synthetic data," *International Journal of Computer Vision*, pp. 1–21, 2020.

[43] E. Bart and S. Ullman, "Cross-generalization: Learning novel classes from a single example by feature replacement," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 672–679.

[44] L. Fei-Fei, "Knowledge transfer in learning to recognize visual objects classes," in *Proceedings of the International Conference on Development and Learning*, 2006, p. 11.

[45] M. Fink, "Object classification from a single example utilizing class relevance metrics," in *Proceedings of the Advances in neural information processing systems*, 2005, pp. 449–456.

[46] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Proceedings of the Advances in neural information processing systems*, 2016, pp. 3630–3638.

[47] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proceedings of the Advances in neural information processing systems*, 2017, pp. 4077–4087.

[48] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," 2016.

[49] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume*. JMLR. org, 2017, pp. 1126–1135.

[50] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[51] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proceedings of the International conference on machine learning*, 2016, pp. 1842–1850.

[52] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," *arXiv preprint arXiv:1707.03141*, 2017.

[53] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *ECCV Workshops*. Springer International Publishing, 2016, pp. 850–865.

[54] L. Bertinetto, J. a. F. Henriques, J. Valmadre, P. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 523–531.

[55] X. Dong, J. Shen, W. Wang, L. Shao, H. Ling, and F. Porikli, "Dynamical hyperparameter optimization via deep reinforcement learning in tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[56] X. Dong and J. Shen, "Triplet loss in siamese network for object tracking," in *Proceedings of the European Conference on Computer Vision*, September 2018.

[57] X. Dong, J. Shen, D. Wu, K. Guo, X. Jin, and F. Porikli, "Quadruplet network with one-shot learning for fast visual object tracking," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3516–3527, 2019.

[58] X. Dong, J. Shen, W. Wang, Y. Liu, L. Shao, and F. Porikli, "Hyperparameter optimization for tracking with continuous deep q-learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[59] M. A. Hossain, M. Kumar, M. Hosseinzadeh, O. Chanda, and Y. Wang, "One-shot scene-specific crowd counting."

[60] M. K. K. Reddy, M. Hossain, M. Rochan, and Y. Wang, "Few-shot scene adaptive crowd counting using meta-learning," *arXiv preprint arXiv:2002.00264*, 2020.

[61] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 403–412.

[62] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

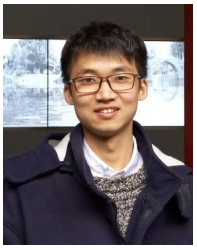
[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[64] 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.

[65] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
[66] 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.
[67] W. Wang, J. Shen, X. Dong, A. Borji, and R. Yang, "Inferring salient objects from human fixations," *IEEE transactions on pattern analysis and machine intelligence*, 2019.
[68] W. Wang, J. Shen, X. Dong, and A. Borji, "Salient object detection driven by fixation prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1711–1720.
[69] W. Wang and J. Shen, "Deep visual attention prediction," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2368–2378, 2017.

Qi Wang (M'15-SM'15) received the B.E. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. He is currently a Professor with the School of Computer Science and with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.

Tao Han received the B.E. degree in transportation equipment and control engineering from the Northwestern Polytechnical University, Xian, China, in 2019. he is currently working toward the M.S. degree in computer science and technology in the Center for OPTical IMagery Analysis and Learning, School of Computer Science, Northwestern Polytechnical University, Xian, China. His research interests include computer vision and pattern recognition.



Junyu Gao received the B.E. degree in computer science and technology from the Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China, in 2015. He is currently pursuing the Ph.D. degree from Center for Optical Imagery Analysis and Learning, Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.

Yuan Yuan (M'05-SM'09) is currently a Full Professor with the School of Computer Science and the Center for Optical Imagery Analysis and Learning, Northwestern Polytechnical University, Xi'an, China. She has authored or co-authored over 150 papers, including about 100 in reputable journals, such as the IEEE TRANSACTIONS AND PATTERN RECOGNITION, as well as the conference papers in CVPR, BMVC, ICIP, and ICASSP. Her current research interests include visual information processing and image/video content analysis.