Dynamic Momentum Adaptation for Zero-Shot Cross-Domain Crowd Counting

Qiangqiang Wu
Department of Computer Science, City University of Hong Kong
qiangqwu2-c@my.cityu.edu.hk

Jia Wan
Department of Computer Science, City University of Hong Kong
jiawan1998@gmail.com

Antoni B. Chan*
Department of Computer Science, City University of Hong Kong
abchan@cityu.edu.hk

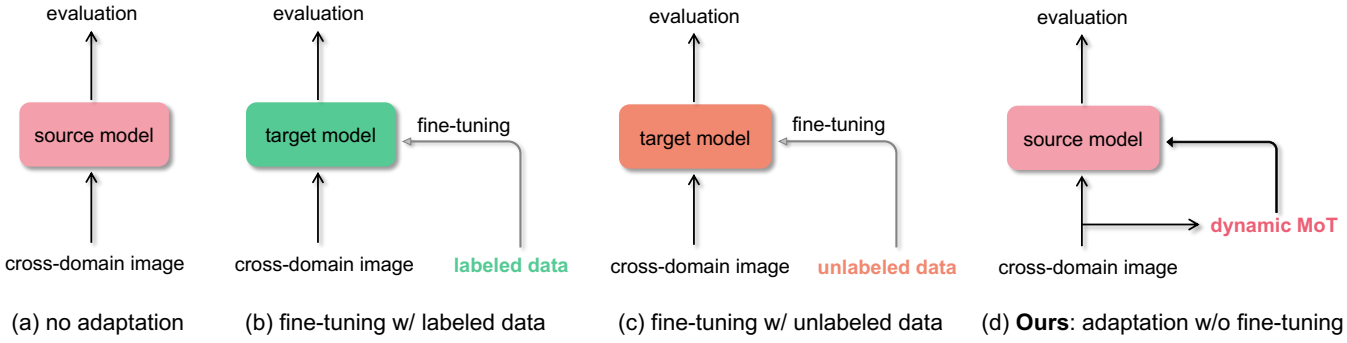

Figure 1: Conceptual comparisons of four mechanisms in the cross-domain crowd counting evaluation. (a) Directly applying a counting model trained in a source domain for cross-domain crowd counting evaluation. (b) Fine-tuning the source model with labeled target data. (c) Fine-tuning the source model with large-scale unlabeled target data. (d) Ours: encoding domain specific information via the proposed dynamic Momentum Template (MoT). Without using any domain specific data for fine-tuning, our method can achieve leading zero-shot cross-domain crowd counting performance.

ABSTRACT

Zero-shot cross-domain crowd counting is a challenging task where a crowd counting model is trained on a source domain (i.e., training dataset) and no additional labeled or unlabeled data is available for fine-tuning the model when testing on an unseen target domain (i.e., a different testing dataset). The generalization performance of existing crowd counting methods is typically limited due to the large gap between source and target domains. Here, we propose a novel Crowd Counting framework built upon an external Momentum Template, termed C^2MoT, which enables the encoding of domain specific information via an external template representation. Specifically, the Momentum Template (MoT) is learned in a momentum updating way during offline training, and then is dynamically updated for each test image in online cross-dataset evaluation. Thanks to the dynamically updated MoT, our C^2MoT effectively generates dense target correspondences that explicitly accounts for head regions, and then effectively predicts the density map based on the normalized correspondence map. Experiments on large scale datasets show that our proposed C^2MoT achieves leading zero-shot cross-domain crowd counting performance without model fine-tuning, while also outperforming domain adaptation methods that use fine-tuning on target domain data. Moreover, C^2MoT also obtains state-of-the-art counting performance on the source domain.

CCS CONCEPTS

• **Information systems** → **Multimedia information systems**; • **Human-centered computing** → **Collaborative and social computing**; **Visualization**; • **Computing methodologies** → **Computer vision**.

KEYWORDS

Zero-Shot Cross-Domain Crowd Counting, Momentum Template, Domain Adaptation

ACM Reference Format:
Qiangqiang Wu, Jia Wan, and Antoni B. Chan. 2021. Dynamic Momentum Adaptation for Zero-Shot Cross-Domain Crowd Counting. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475230

1 INTRODUCTION

Crowd counting is the task of automatically counting the total number of people in surveillance images or videos. This task is an essential research topic and has gained much attention in both academic and industrial fields.

Recent advances [39, 50] in crowding counting have witnessed significant progress, with existing counting methods achieving promising performance on large-scale crowd counting datasets, where the crowd counter is trained and tested on the same dataset (same domain). However, we argue that this training and evaluation

*Corresponding author.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475230

scheme still has a large gap to practical usage, where the crowd counters are required to perform favourably on both similar surveillance crowd scenes (source domain), as well as generalize to unseen crowd scenes and crowd distributions (i.e., the target domain). One naive solution is to directly evaluate the counter trained on the specific dataset on various test datasets with variety of unseen scenes. This however typically leads to large performance drop due to the domain gap.

To alleviate the domain gap problem, several works have investigated leveraging labeled or large-scale unlabeled data to fine-tune the counters on the target domain for domain adaptation. As a representative work, [8] performs model fine-tuning with a photo-realistic dataset that is generated by applying style transfer on a large-scale synthetic dataset. [49] proposes a method based on the fine-tuning on a re-sampled dataset. However, online fine-tuning is time-consuming. Moreover, in real-world applications, labeled data in the target domain may not always be available for fine-tuning, and large-scale unlabeled data in the target domain may also be unavailable (e.g., for a new camera installation). These limitations affect the practicality of real-world crowd counting applications in unseen scenes.

To address the above problem, we introduce a new task that is closer to real-world applications, termed *zero-shot cross-domain crowd counting*. We consider the case where no additional labeled or unlabeled data is available for model fine-tuning. Given a single test image without any annotations in the target domain, a counter trained in the source domain is directly employed to predict the total count in the test image. In this zero-shot cross-domain evaluation setting, most existing domain adaptation-based methods [8, 46, 49] cannot work since there is no domain specific training data for model fine-tuning. Moreover, existing counting methods suffer a large performance drop in this zero-shot cross-domain evaluation, due to the lack of online adaptation to bridge the domain gap.

In this paper, we propose a novel crowd counting framework for *zero-shot cross-domain crowd counting*. As illustrated in Fig. 1, different from traditional crowd counting frameworks that implicitly encode domain information by training a counting model, our proposed crowd counting framework, denoted as C^2MoT, is based on an external momentum template (MoT), which explicitly encodes domain specific information. The MoT enables the model to perform domain adaptation in a single feed-forward without model fine-tuning. Specifically, the proposed MoT is firstly learned in a momentum updating way during offline training in the source domain, and then is dynamically updated for each test image in the target domain. The basic idea is to aggregate reliable online target (i.e., head) features for the online updating of MoT. By treating the updated MoT as the target kernel, our C^2MoT can generate accurate dense head correspondences that explicitly account for head regions in the test image via a cross-correlation operation. The density map is then predicted from this normalized cross-correlation map.

We conduct experiments on four large-scale crowd counting datasets, including ShanghaiTech [50], UCF-QNRF [13], NWPU-Crowd [44] and JHU-CROWD++ [37]. Our proposed C^2MoT achieves leading zero-shot cross-domain crowd counting performance without model fine-tuning, and also outperforms domain adaptation methods that use target domain fine-tuning. In addition to the zero-shot cross-domain evaluation, we also evaluate C^2MoT using the normal source domain evaluation, where our C^2MoT also obtains state-of-the-art counting performance. In summary, our main contributions are:

- We propose a novel crowd counting framework built upon an external Momentum Template (MoT), which explicitly encodes domain specific information for domain adaptation without model fine-tuning.
- We propose a dynamic momentum adaptation method, which dynamically updates the MoT for each specific test image in unseen crowd scenes. By leveraging the dynamically updated MoT, we can effectively find dense head correspondences in unseen scenes, which are useful for the final density map prediction.
- Extensive experiments on four large-scale crowd counting datasets demonstrate that our proposed method achieves state-of-the-art counting performance on both zero-shot cross-domain evaluation (also outperforming domain adaptation methods that use fine-tuning on target domain data) and standard source domain evaluation.

2 RELATED WORK

In this section, we review the relevant work about crowd counting and domain adaptation approaches.

2.1 Crowd Counting

Traditional crowd counting algorithms count the number of people via either detection [9] or direct regression [3] using low-level features [4, 12]. [9] proposes to count the crowd number based on the detection of the whole human body, while [17] is based on the detection of body parts. To avoid detection, [3] proposes to directly estimate the crowd count based on the low-level features extracted from the image.

Recent approaches focus on using deep neural networks to estimate crowd density maps, which are smoothed heatmaps generated from the dot annotations. Different architectures [50] and loss functions [39] are proposed to deal with challenges such as scale variation and annotation noise [39]. A multi-column neural network [50] is proposed to extract multi-scale features using a network with multiple branches with different receptive field. [34] propose to select a proper branch instead of fusing the multi-scale features. [2] proposes a scale aggregation architecture to model multi-scale information for each layer. [15] utilized an image pyramid to handle scale variations. [25] proposes a context-aware method to encode contextual information. To predict high-quality density maps, [31] proposes a two-stage refinement approach, while [32] proposes a feedback mechanism, and [23] proposes a region-based refinement method.

Most of the density map based approaches adopt pixel-wise mean squared error (MSE) as the loss function. However, the choice of Gaussian bandwidth used to generate the density map affects the performance. Thus, [38, 42] propose to learn the density map supervision for a given dataset and architecture. [29] proposes a point-wise Bayesian Loss (BL), which directly uses dot annotation as the supervision. [39] proposed a robust loss function to model the noisy point annotations, while [40] proposes a generalized loss function and proves that both MSE and BL are special cases of it.

The above previous approaches mainly focus on training and testing counting models on a specific source domain, and their performances drop significantly under cross-domain evaluation due to the lack of domain adaptation. In contrast, our proposed method is also only trained on the source domain, but achieves significantly better generalization performance when tested on target domains due to our external adapted dynamic MoT.

2.2 Domain Adaptation

For cross-domain crowd counting, fine-tuning based approaches are proposed to apply crowd counting to novel scenes. [49] proposes fine-tuning a pre-trained crowd counting model on a re-sampled dataset. [45] proposes a Generative Adversarial Network (GAN) to adapt the model trained on synthetic data to real-world images. [46] proposes a neuron transformation to model domain shifts. [41] proposes to model the residual between two samples to improve the generalization ability, but the support images used for training is fixed, which limits the adaptation ability. [8] proposes to extract domain-invariant features via adversarial training. In [30], multiple domain specific modules are trained using labeled data from target domains for online switching. [27] proposes to take advantage of both detection and regression-based counting frameworks, and fine-tunes the offline learned counter via online-estimated pseudo labels.

The above cross-domain crowd counting methods require complex fine-tuning based on re-sampled dataset or unlabeled/labeled data, which requires additional online optimization steps for different scenes. Thus, methods that require fine-tuning may not be practical for quick deployment of surveillance systems. Moreover, most of the cross-domain methods only model the appearance shift with a GAN, while the shift of crowd density distribution is not considered. To address those issues, we propose a zero-shot adaptation method, which can be directly applied to different domains without fine-tuning the model. We also propose a dynamic momentum template to model the shift in the crowd density distribution. It should be noted that our proposed zero-shot domain adaptation framework is generic, and it can be applied to domain adaptation or transfer in other tasks (e.g., visual tracking [1] and segmentation [28]) by modelling source domain knowledge via external representation (offline MoT) and then generating an external dynamic MoT for each specific online testing task.

3 OUR APPROACH

In this section, we first revisit the basic crowd counting framework and illustrate its limitation in the zero-shot cross-domain crowd counting task. We then introduce a novel crowd counting framework based on an external Momentum Template (MoT) representation. We finally show how to dynamically update the MoT for each specific test image in unseen crowd scenes in the target domain.

3.1 Basic Crowd Counting Framework

In the basic crowd counting framework, a crowd counting model is typically trained on a training set to encode source domain information via offline model updating, and then tested on a test set within the same dataset. The main assumption behind this training and testing scheme is that the data distribution of the training set should be similar to the distribution of the test set (i.e., both sets are from the source domain), such that the source crowd counting model can generalize well to the unseen test set. The basic counting framework can be implemented with various designs, e.g., total count prediction [49] and density map estimation [42].

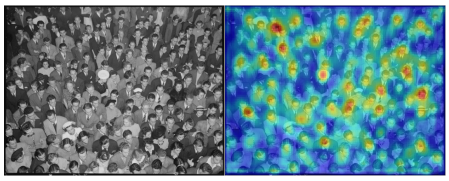

Figure 2: Left: The input image. Right: The cross-correlation map generated via a mean target vector calculated based on the initial memory bank built on SHA.

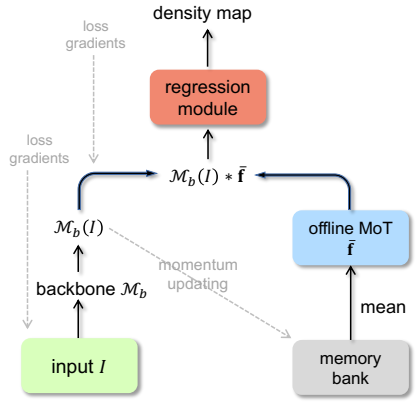

Figure 3: Crowd counting framework using offline Momentum Template (MoT). $*$ is a depth-wise convolution operation. The solid lines indicate the inference stage, and the dashed lines show the updating of the counting model and the memory bank during the training stage.

Formally, given training images I in the source domain (i.e., training set) and a crowd counting model $\mathcal{M}$, the basic crowd counting framework trains the model by minimizing a counting loss $\mathcal{L}(\mathcal{M}(I), Y)$, where Y denotes the ground-truth, which can be a count number, a density map or an annotated dot map. The loss function $\mathcal{L}$ may have various formulations, e.g., the mean square error (MSE) loss [42] with Y as a groundtruth density map, and Bayesian Loss (BL) [29] with Y as a groundtruth dot map.

Despite the great success achieved by the basic crowd counting framework based methods in the source domain evaluation, their performances drop significantly in the cross-domain evaluation due to the large domain gap (as shown in Fig. 5b). Since the basic counting framework encode domain specific information in the model parameters, one basic idea to improve the generalization performance is to finetune the source counting model in the target domain by leveraging labeled or large-scale unlabeled target data. However, the target domain data is not always available in the real-world applications (e.g., deployment of a counting system in a temporary event like an outdoor concert). This may severely limit the real-world applications of existing crowd counting methods in the zero-shot cross-domain crowd counting problem.

In order to address the above problem, we propose a novel crowd counting framework built upon an external momentum template, which explicitly encodes domain specific information via an external representation, instead of within the model parameters as

with the traditional approach. This new framework enables effective adaptation to a new target domain without model fine-tuning. In the next subsection, we will give a detailed description of our proposed novel crowd counting framework.

3.2 Momentum Template Representation

In this section, we propose a novel crowd counting framework built upon an external Momentum Template (MoT) for zero-shot cross-domain crowd counting. The proposed MoT encodes rich target head information, and is calculated using a momentum-updated memory bank. Our hypothesis is that an effective MoT can accurately find target head correspondences in a test image by using a cross-correlation operation, and the resulting target correspondence map can be used to predict the density map. For this MoT guided crowd counting framework, we will introduce: 1) building of the memory bank; 2) momentum updating of the memory bank; 3) calculation of MoT based on the memory bank, and 4) how to perform crowd counting with the MoT by using a cross-correlation operation.

Building the Memory Bank. The first step is to build a memory bank. Since only the source domain data is available for this zero-shot cross-domain crowd counting problem, the memory bank is built based on the source domain training data. Given a source domain training image I_s with ground-truth head annotations $\{(x_i, y_i)\}_{i=1}^{N_s}$, we first use a backbone feature extractor $\mathcal{M}_b$ to extract convolutional feature map of I_s. For each annotation position (x_i, y_i) in the input space, the corresponding location at the extracted backbone feature map can be calculated as $(\frac{x_i}{\Delta} - 0.5, \frac{y_i}{\Delta} - 0.5)$, where Δ is the total stride of the backbone feature extractor. The feature at the annotation point $(\frac{x_i}{\Delta} - 0.5, \frac{y_i}{\Delta} - 0.5)$ is extracted via a RoIAlign operation with the output size of 1×1. Thus for the i-th head point annotation, a feature vector $f_i \in \mathbb{R}^D$ is extracted. Storing all the head point features in the memory bank is not memory efficient. Here, we calculate a mean head feature for each training image, and thus obtain an image-level feature set $\mathcal{F} = \{\bar{\mathbf{f}}_j\}_{j=1}^{N}$, where N is the number of images. The memory bank is firstly initialized to store the whole set $\mathcal{F}$ and their corresponding keys (i.e., image names). Note that similar to existing methods [29, 42], we use a pre-trained VGG19 backbone network as $\mathcal{M}_b$ for feature extraction. The role of the memory bank is to store the specific appearance features of the heads in each image. In Fig. 2, we show that a mean target vector calculated based on the initial memory bank can effectively generate target head correspondences in a testing image via a cross-correlation operation.

Momentum Updating. After obtaining the initial memory bank, we update its contents during offline training of the counting model. Specifically, as shown in Fig. 3, for each training iteration, we firstly calculate the MoT $\bar{\mathbf{f}} = \frac{1}{N}\sum_{j=1}^{N} \bar{\mathbf{f}}_j$, which averages all the image-level feature vectors $\mathcal{F}$ stored in the memory bank. The MoT and the feature map extracted from the input image are further input to a depth-wise cross correlation layer, i.e., treating $\bar{\mathbf{f}}$ as a target kernel and performing depth-wise convolution on the input feature map $\mathcal{M}_b(I)$. The obtained cross-correlation map is then used for density map prediction:

$$\mathbf{m} = \mathcal{M}_r(LN(\bar{\mathbf{f}} * \mathcal{M}_b(I))), \tag{1}$$

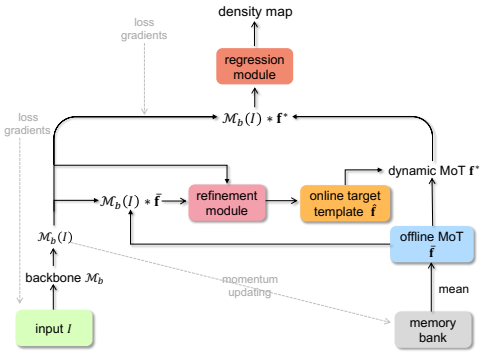

Figure 4: Crowd counting framework using dynamic MoT. $*$ is a depth-wise convolution operation. The solid lines indicate the inference stage, and the dashed lines show the updating of the counting model and the memory bank during the training stage.

where $\mathbf{m}$ is a density map, $*$ indicates a depth-wise convolution operation, $LN(\cdot)$ is a layer normalization operation that can alleviate the affect of response-value shift during cross-domain evaluation, and $\mathcal{M}_r$ represents a density regression module. For offline training, we use the Bayesian counting framework [29] as the basic counting framework. For fair comparison, we use the same architecture for the density prediction module $\mathcal{M}_r$, backbone feature extractor $\mathcal{M}_b$, and Bayesian loss for supervision as in [29].

With the updating of $\mathcal{M}_b$ during training, the target feature set $\{\bar{\mathbf{f}}_j\}_{j=1}^{N}$ gradually becomes outdated. For each training iteration, updating the whole target feature set is time-consuming. Here, we propose to update the feature vectors of the sampled mini-batch images in a momentum updating way:

$$\bar{\mathbf{f}}_j = (1 - r)\bar{\mathbf{f}}_j + r\bar{\mathbf{f}}_j^{'}, \tag{2}$$

where $r \in [0, 1]$ is a momentum coefficient and $\bar{\mathbf{f}}_j^{'}$ indicates the target mean vector of the j-th training image calculated with the updated $\mathcal{M}_b$. The updated $\bar{\mathbf{f}}_j$ is further stored into the memory bank.

The proposed MoT can explicitly encode domain-specific information of the *training set* via its external representation. In the next subsection, we show how to learn to dynamically update the MoT on each specific testing image for domain adaptation.

3.3 Dynamic Momentum Adaptation

Given a testing image in a new target domain, the MoT is adapted by aggregating reliable online target head information in the testing image. The goal is to encode domain specific target information, such that the adapted MoT generates more accurate correspondence (cross-correlation) maps for the density map predictor. The pipeline is shown in Fig. 4. We firstly use the offline MoT calculated on the memory bank to generate a weight map, which represents the potential head locations in the test image. We then calculate an online target template based on the generated weight map and the input testing image. Finally, the dynamic MoT is computed by combining both the offline MoT and online target template, which is then used for density map prediction. We next describe these steps in detail.

By treating the offline MoT $\bar{\mathbf{f}}$ as a convolutional kernel, we firstly use a cross-correlation operation to generate a multi-channel correspondence map $\mathbf{V} = \bar{\mathbf{f}} * \mathcal{M}_b(I) \in \mathbb{R}^{H \times W \times D}$, which indicates

Table 1: Evaluation of cross-domain performance of crowd counting: (top) no adaptation; (middle) domain adaptation via fine-tuning on target domain data; (bottom) domain adaptation without fine-tuning, i.e., zero-shot cross-domain. 'A', 'B' and 'Q' refer to SHA, SHB and UCF-QNRF. *Syn* indicates a large-scale source synthetic dataset (i.e., GCC [45]) used for transfer learning. *Real* denotes a real source dataset (i.e., SHA or SHB) for source domain training. The best performance is highlighted bold, and 2nd best is underlined.

	Method			Fine-tuning	Adaptation	A → B		A → Q		B → A		B → Q	
						MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
no adaptation	MCNN [50]	2016	*Real*	✗	✗	85.2	142.3.	-	-	221.4	357.8	-	-
	D-ConvNet [35]	2018	*Real*	✗	✗	49.1	99.2	-	-	140.4	226.1	-	-
	SPN+L2SM [48]	2019	*Real*	✗	✗	21.2	38.7	227.2	405.2	126.8	203.9	-	-
	RegNet [21]	2019	*Real*	✗	✗	21.7	37.6	198.7	329.4	148.9	273.9	267.3	477.6
	DetNet [24]	2019	*Real*	✗	✗	55.5	90.0	411.7	731.4	242.8	400.9	411.7	731.4
	D2CNet [6]	2021	*Real*	✗	✗	21.6	34.6	126.8	245.5	164.5	286.4	267.5	486.0
adaptation w/ fine-tuning	Cycle GAN [51]	2017	*Syn*	✓	✓	25.4	39.7	257.3	400.6	143.3	204.3	257.3	400.6
	SE CycleGAN [45]	2019	*Syn*	✓	✓	19.9	28.3.	230.4	384.5	123.4	193.4	230.4	384.5
	SE+FD [10]	2020	*Syn*	✓	✓	16.9	24.7	221.2	390.2	129.3	**187.6**	221.2	390.2
	RBT [27]	2020	*Real*	✓	✓	13.4	29.3	175.0	294.8	**112.2**	218.2	211.3	381.9
zero shot adaptation	BL [29] (baseline)	2019	*Real*	✗	✗	15.9	25.8	166.7	287.6	138.1	228.1	226.4	411.0
	C^2MoT	**Ours**	*Real*	✗	✓	**12.4**	**21.1**	**125.7**	**218.3**	120.7	192.0	**198.9**	**368.0**

potential target head positions in the input image I. To further refine the correspondence map, we propose refinement module (RM) formulated as:

$$\mathbf{W} = \text{softmax}(F([\Phi_A(\mathbf{V}), \Phi_M(\mathbf{V})])), \tag{3}$$

where $\Phi_A(\cdot)$ and $\Phi_M(\cdot)$ respectively are average and max pooling operations, $\Phi_A(\mathbf{V}) \in \mathbb{R}^{H\times W}$, and $F(\cdot)$ is a convolutional operation with the kernel size of 7×7 and the output dimension of 1. The output of the RM is a weight map $\mathbf{W} \in \mathbb{R}^{H\times W}$, where each position represents the probability that the corresponding position is a target head, and the of $\mathbf{W}$ equals to 1.

With the weight map $\mathbf{W}$, we calculate an online target template:

$$\hat{\mathbf{f}} = \mathbf{W} \times \mathcal{M}_b(I), \tag{4}$$

where we reshape the $\mathbf{W}$ and the extracted feature map ($\mathcal{M}_b(I)$) of the input image to $\mathbf{W} \in \mathbb{R}^{1\times HW}$ and $\mathcal{M}_b(I_s) \in \mathbb{R}^{HW\times D}$, respectively. We finally get an online target template $\hat{\mathbf{f}} \in \mathbb{R}^{1\times D}$ with the above matrix multiplication operation. Finally, the dynamic MoT is formed by combining the offline MoT $\bar{\mathbf{f}}$ and online target template $\hat{\mathbf{f}}$:

$$\mathbf{f}^* = \frac{1}{1+e^{-\alpha}}\bar{\mathbf{f}} + \frac{e^{-\alpha}}{1+e^{-\alpha}}\hat{\mathbf{f}}, \tag{5}$$

where α is learnable parameter that is initialized to zero in the start of training. The dynamic MoT contains the source information (the offline MoT $\bar{\mathbf{f}}$) that is adapted to the target domain via the online target template $\hat{\mathbf{f}}$.

Compared with several similar works [5, 47] in image classification, there are several main differences in our RM design: 1) our RM is built on the top of the cross-correlation map instead of appearance features used in [29, 42]. Since the final output of RM is a probabilistic weight map, the input cross-correlation map is closer to the final probabilistic weight map, which makes the refinement easier to be achieved. 2) Our RM models correlations between locations in the correspondence map with a SoftMax function.

Training. During the offline training stage, each training image is considered as a new target domain and our method is trained to learn to generate a dynamic MoT for domain adaptation. In particular, as illustrated in Fig. 4, a dynamic MoT $\mathbf{f}^*$ that encodes specific online target information is firstly generated from the input image and offline MoT. Then we use the dynamic MoT $\mathbf{f}^*$ instead of the offline MoT in (1) for predicting the density map. Using the same steps described in Section 3.2, we then calculate a Bayesian loss for end-to-end training. Here, for each training iteration, the backbone feature extractor $\mathcal{M}_b$, the density regression module $\mathcal{M}_r$, the convolutional kernel F and the parameter α are updated. Finally, the memory bank and offline MoT are updated in each training iteration with the updating of $\mathcal{M}_b$ based on (2).

4 EXPERIMENTS

In this section, we evaluate the zero-shot crowd-domain crowd counting performance of our proposed C^2MoT and current state-of-the-art counting methods.

4.1 Experiment setup

Datasets. We evaluate our method on four datasets including ShanghaiTech [50], UCF-QNRF [13], NWPU-Crowd [44] and JHU-CROWD++ [37]. ShanghaiTech consists of two parts: SHA and SHB. SHA and SHB respectively contain 300/400 training images and 182/316 testing images. UCF-QNRF contains 1535 high resolution images with 1,201 for training and 334 for testing. JHU-CROWD++ and NWPU-CROWD are two recently proposed large-scale datasets. For JHU-CROWD++, it contains 4,317 high-resolution images in total, with 2,722/500/1,600 images for training/validation/testing. For NWPU-CROWD, there are 3,109/500/1,500 training/validation/testing images. Note that the annotations of the testing images in NWPU-CROWD are not released to the public, and the test performance of our method is obtained via the official online evaluation server.

Evaluation protocol. For zero-shot cross-domain crowd counting, we train our model on the training set of the source domain (e.g., SHA), and evaluate it directly on the test sets of the other datasets (e.g., SHB, UCF-QNRF, NWPU-CROWD, and JHU-CROWD++). We use the widely used metrics, mean absolute error (MAE) and mean

Table 2: Performance of zero-shot cross-domain counting using NWPU-CROWD as the source domain and other datasets as the target domain. All models are trained on the source training data, are directly tested on the target domain without fine-tuning. The number of testing images in each dataset is shown in brackets. "Overall" indicates the performance on the combined test sets of the source and target domains. The best performance is in bold, and 2nd best is underlined.

Method		ShanghaiTech (498)		UCF-QNRF (334)		JHU-Val (500)		JHU-Test (1,600)		Overall (4,432)	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MCNN [50]	2016	79.3	139.4	308.7	502.4	138.3	441.9	163.5	485.3	185.5	548.7
CSRNet [18]	2018	49.2	103.6	159.0	287.9	111.3	248.5	130.6	349.4	118.3	330.7
C3F-VGG [7]	2019	43.3	93.1	171.5	331.9	93.1	**230.4**	111.2	333.4	-	-
SCAR [21]	2019	42.3	98.6	148.8	251.2	129.4	304.5	144.5	408.5	-	-
BL [29]	2019	34.9	83.1	104.1	208.4	86.8	294.7	92.0	318.1.	90.4	346.7
DM [43]	2020	32.9	73.7	101.2	201.3	88.1	248.7	96.3	308.1	85.9	297.0
NoiseCC [39]	2020	37.4	75.3	109.2	188.2	84.5	269.6	93.4	314.3	88.5	379.2
D2CNet [6]	2021	41.7	101.8	110.3	187.2	-	-	-	-	-	-
C^2MoT	**Ours**	**31.7**	**69.1**	**94.8**	**177.2**	**82.4**	281.7	**81.8**	**288.8**	**76.6**	**292.9**

Table 3: Performance of zero-shot cross-domain counting using JHU-CROWD++ as the source domain and other datasets as the target domain. All models are trained on the source training data, are directly tested on the target domain without fine-tuning.

Method		ShanghaiTech (498)		UCF-QNRF (334)		NWPU-Val (500)		NWPU-Test (1,500)		Overall (4,432)	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
BL [29]	2019	56.0	117.5	190.8	332.9	114.1	521.9	133.0	478.9	105.6	388.2
DM [43]	2020	47.4	108.0	161.4	301.3	99.1	**471.4**	129.4	500.6	97.2	383.4
NoiseCC [39]	2020	43.3	109.1	139.9	259.9	95.5	527.8	112.6	443.6	88.7	358.6
D2CNet [6]	2021	49.9	108.0	171.4	302.2	-	-	-	-		
C^2MoT	**Ours**	**43.1**	**101.5**	**135.3**	**251.9**	**89.4**	499.5	**110.5**	**426.3**	**84.1**	**344.8**

squared error (MSE), to measure counting performance:

$$\text{MAE} = \frac{1}{N}\sum_i |g_i - \hat{g}_i|, \quad \text{MSE} = \sqrt{\frac{1}{N}\sum_i |g_i - \hat{g}_i|^2}, \qquad (6)$$

where N is image numbers, g and $\hat{g}$ indicate GT and predicted counts.

Implementation details. For fair comparison, we use the same backbone feature extractor $\mathcal{M}_b$ (i.e., VGG19 pre-trained on ImageNet), density prediction module $\mathcal{M}_r$ and Bayesian loss as used in [29]. The momentum coefficient r in (2) is set to 0.5. The parameter α in (5) is initialized to zero and learned during end-to-end offline training. We use the Adam optimizer [16] with the learning rate of 10^{-5} for end-to-end training.

4.2 Zero-Shot Cross-Domain Crowd Counting

Source domain training on ShanghaiTech. Following the experiment design of other domain adaptation methods [10, 27], we first evaluate cross-domain counting using SHA or SHB as the source domain, and SHB/SHA and UCF-QNRF as the target domains. We compare our proposed C^2MoT with two main types of counting methods: 1) fine-tuning based domain adaptation methods, including Cycle GAN [51], SE CycleGAN [45], SE+FD [10] and RBT [27]; 2) state-of-the-art counting methods without domain adaptation [6, 21, 24, 29, 35, 48, 50]. For Cycle GAN, SE CycleGAN and SE+FD, a large-scale synthetic dataset GCC [45] is used together with the target domain data for fine-tuning via adversarial learning.

The experiment results are presented in Table 1. Despite not using model fine-tuning, our C^2MoT still achieves the best overall transfer performance on all the four cross-dataset transfer experiments (in terms of MAE, ranking 1st on 3 transfer experiments, and 2nd on 1 experiment). Specifically, our C^2MoT achieves the best performance on transfer experiments $\mathbf{A} \rightarrow \mathbf{B}$, $\mathbf{A} \rightarrow \mathbf{Q}$ and $\mathbf{B} \rightarrow \mathbf{Q}$, which demonstrates that our dynamic MoT can bridge the domain gap and transfer well to more congested scenes (i.e., UCF-QNRF). In contrast, on the transfer experiment $\mathbf{B} \rightarrow \mathbf{A}$, our method is inferior to RBT. The main reason is that there are many similar scenes in SHB and SHA, and thus fine-tuning based methods can benefit more from this property by leveraging the similar target domain data for fine-tuning. Note that our method can achieve the best transfer performance on $\mathbf{A} \rightarrow \mathbf{B}$. This is because there are more head annotations in SHA, which can provide a more informative offline MoT for better generalization.

Compared with state-of-the-art counting methods without domain adaptation, our method achieves the best transfer performance among these methods and significantly outperforms the baseline BL by large margins. Note that our method uses the same evaluation setup as these state-of-the-art methods, training only on the source domain data, which shows the effectiveness of the proposed dynamic MoT for domain adaptation.

Source domain training on large-scale datasets. We next evaluate the zero-shot cross-domain performance using a large-scale dataset as the source domain, and the other datasets as the target domain. Specifically, we train our method and other state-of-the-art counting methods on the training set of NWPU-CROWD, and directly evaluate the models on the test sets of the other datasets, i.e., without any fine-tuning on target data. Here we do not compare fine-tuning based methods because: 1) additional domain specific data is used for fine-tuning in these methods, which violates the zero-shot requirement; 2) their models are unavailable on large-scale datasets for comparison.

Table 2 reports the experiment results using NWPU-CROWD as the source domain. Our method achieves the leading transfer performance on ShanghaiTech (i.e., combining both SHA and SHB), UCF-QNRF and both validation and testing sets of JHU-CROWD++ Our method works well at domain adaptation even on UCF-QNRF and JHU-test datasets, which have challenging congested scenes and large count ranges. To measure the overall counting performance, we also report the overall MAE and MSE performance on the test data of both the source and target domains. Our C^2MoT achieves the best MAE/MSE overall performance (76.6/292.9) among the state-of-the-art methods. Compared with the baseline method

Table 4: Source domain comparison with state-of-the-art crowd counting methods on large-scale datasets.

		NWPU-CROWD		JHU-CROWD++		UCF-QNRF	
		MAE	MSE	MAE	MSE	MAE	MSE
MCNN [50]	CVPR'16	232.5	714.6	188.9	483.4	277.0	426.0
SwitchCNN [34]	CVPR'17	-	-	-	-	228.0	445.0
CSRNet [19]	CVPR'18	121.3	387.8	85.9	309.2	110.6	190.1
CL [13]	ECCV'18	-	-	-	-	132.0	191.0
SANet [2]	ECCV'18	190.6	491.4	91.1	320.4	-	-
DSSINet [22]	ICCV'19	-	-	133.5	416.5	99.1	159.2
MBTTBF [36]	ICCV'19	-	-	81.8	299.1	97.5	165.2
BL [29]	ICCV'19	105.4	454.2	75.0	299.9	88.7	154.8
LSCCNN [33]	TPAMI'20	-	-	112.7	454.4	120.5	218.2
KDMG [42]	TPAMI'20	100.5	415.5	69.7	268.3	99.5	173.0
ASNet [14]	CVPR'20	-	-	-	-	91.6	159.7
AMSNet [11]	ECCV'20	-	-	-	-	101.8	163.2
AMRNet [26]	ECCV'20	-	-	-	-	86.6	152.2
LibraNet [20]	ECCV'20	-	-	-	-	88.1	**143.7**
DM-count [43]	NeurIPS'20	88.4	**357.6**	68.4	283.3	85.6	148.3
NoiseCC [39]	NeurIPS'20	96.9	534.2	67.7	258.5	85.8	150.6
D2CNet [6]	TIP'21	85.5	361.5	73.7	292.5	84.8	145.6
Ours		**79.9**	360.0	**59.7**	**254.2**	**80.7**	**143.7**

BL, our C^{2}MoT gains large improvements in terms of the MAE and MSE metrics on all the datasets. Note that C^{2}MoT uses the same Bayesian loss [29] and has the similar model capacity with BL, which further demonstrates the advantages of our proposed dynamic MoT for zero-shot domain adaptation.

We also tested using JHU-CROWD++ as the source domain, and the experiment results are presented in Table 3. Similar to using NWPU-CROWD as the source domain, our C^{2}MoT achieves favorable transfer performance on the target domain datasets, as well as overall performance. We note that using NWPU-CROWD as the source domain leads to lower counting errors in the target domains (e.g., ShanghaiTech and UCF-QNRF). This is mainly because NWPU-CROWD is more similar to the target datasets, while JHU-CROWD++ contains more test images with different styles under challenging weather conditions, e.g., rain and snow.

4.3 Source Domain Evaluation

In this subsection, following the standard evaluation scheme, where the source domain is used for both training and testing, we report the source domain performance of our proposed method and state-of-the-art methods on three large-scale counting datasets, UCF-QNRF, NWPU-CROWD and JHU-CROWD++.

The source domain evaluation results are presented in Table 4. In terms of MAE, our C^{2}MoT achieves the leading performance on all the three datasets, outperforming recent state-of-the-art methods. Our method significantly outperforms the baseline counting method BL by large margins. Specifically, our C^{2}MoT improves BL from 105.4/75.0/88.7 to 79.9/59.7/80.7. This is mainly because our method treats each test image as a new target domain, and is trained to adapt to each specific test image. Thus, our C^{2}MoT still achieves favorable performance in source domain evaluation, by effectively alleviating the domain gap between training and test sets of the dataset.

Note that we are using the same C^{2}MoT models trained on NWPU-CROWD or JHU-CROWD++ in the transfer experiments (Tables 2 and 3) and the source domain evaluation (Table 4). This demonstrates that our method has good generalization, achieving leading zero-shot cross-domain performance as well as state-of-the-art performance using standard source domain evaluation.

Table 5: Ablation study on α. The source domain is JHU-CROWD++ and target domain is UCF-QNRF.

MoT	JHU-CROWD++		UCF-QNRF	
	MAE	MSE	MAE	MSE
only offline MoT ($\alpha \to +\infty$)	61.22	257.18	147.03	270.21
only online target template ($\alpha \to -\infty$)	63.17	263.31	140.95	261.64
learnable α	**59.7**	**254.2**	**135.25**	**251.91**

Table 6: Ablation study on the layer normalization (LN). The source domain is JHU-CROWD++ and target domain is UCF-QNRF.

MoT	JHU-CROWD++		UCF-QNRF	
	MAE	MSE	MAE	MSE
w/o LN	61.67	257.24	159.27	273.76
w/ LN	**59.7**	**254.2**	**135.25**	**251.91**

Table 7: Ablation study on the average and max pooling operations used in the RM module. The performance is evaluated on JHU-CROWD++.

Average Pooling	Max Pooling	MAE	MSE
✓		61.50	260.42
	✓	60.43	259.14
✓	✓	**59.7**	**254.2**

4.4 Ablation Study

In this subsection, we conduct ablation studies on the various components of C^{2}MoT.

Effect of α. α is a parameter used in (5) that controls the weighting between the offline MoT and online target template when calculating the dynamic MoT. Based on (5), when $\alpha \to +\infty$, only the offline MoT is used for crowd counting, while when $\alpha \to -\infty$, only the online target template is used. In our design, we implement α as a learnable parameter, which can be end-to-end learned during offline training. In this ablation study, we train these three variants on the JHU-CROWD++ dataset, and report both the source and target domain performance for comparison in Table 5.

Using a learnable parameter α leads to the best performance on both the source and target domain evaluation, showing the advantage of combining both the offline MoT and online target template. Using only the offline MoT ($\alpha \to \infty$) obtains favorable performance on the source dataset JHU-CROWD++, due to the external offline MoT encoding the source domain information. However, using only the offline MoT has limited generalization performance, due to the lack of online adaptation, which is illustrated by the poor performance on the target domain. In contrast, using only the online target template ($\alpha \to -\infty$) has better generalization performance on the target domain, but inferior performance on the source domain. This is because the online target template could be noisy due to incorrect prediction of the weight map, which may generate an unreliable dynamic MoT for crowd counting. By combining both the offline MoT and online target template with α, the optimal performance can be achieved. The learned α was 0.3256 in this experiment.

Effect of layer normalization. We next study the effect of using layer normalization (LN) in our density map predictor. When evaluating on the target domain, the correspondence map might have large response value shifts, which affects the final density map prediction. LN provides a more stable normalized correspondence map, enabling the better density map estimation when there is domain shift. Table 6 shows the performance on variants with and without using the LN. The variant without LN obtains similar source domain evaluation compared to the variant with LN. However, on the

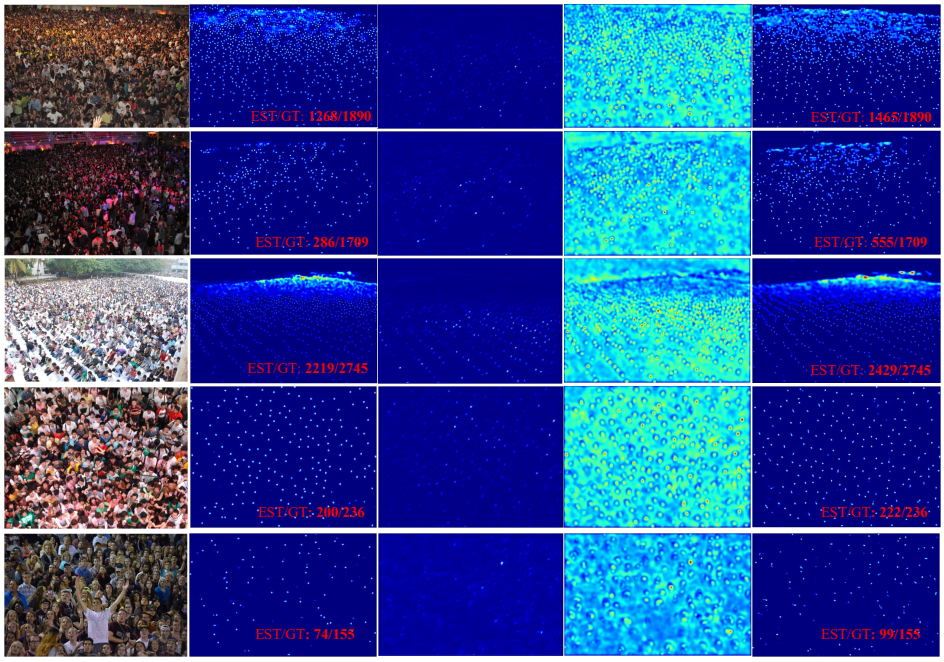

Figure 5: Visualization of the weight map (c), the cross-correlation map (d) generated via the dynamic MoT and density maps estimated by (b) and (e). The input images are sampled from UCF-QNRF [13] while our method and BL are trained on SHB [50] for zero-shot cross-domain evaluation. The estimated and ground-truth counts (EST/GT) are shown in the bottom right of (b) and (e).

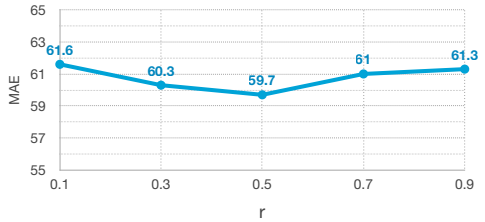

Figure 6: Ablation study on the momentum coefficient r. The performance is evaluated on JHU-CROWD++.

target domain, using LN leads to substantial improvements, which confirms the necessity of LN for cross-domain testing.

Effect of average and max pooling operations in RM. In Table 7, we show that using both average and max pooling operations provides a more informative input for the generation of the weight map in RM. In addition, only using average or max pooling achieves competitive MAE performance due to the reliable initial cross-correlation map generated with the offline MoT.

Effect of r. The momentum coefficient r controls the updating smoothness of the memory bank. As shown in Fig. 6, our method is generally not sensitive to the selection of r. Using too slow (i.e., 0.1) or too fast (i.e., 0.9) updating only causes small performance drop. This is mainly because the updating of the memory bank is jointly conducted with the model training, which makes the two processes adapt to each other in the end-to-end training.

Detailed visualization. We show the detailed visualization of the generated weight map, cross-correlation map and density maps estimated by the baseline BL and our method in Fig. 5. Our method effectively aggregates reliable online target information for the generation of dynamic MoT. For our method, the weight maps assign weights on the head positions for computing the online templates (Fig. 5c), and the cross-correlation maps effectively indicate the head positions (Fig. 5d), which provides a strong prior for the final density map prediction (Fig. 5e). Compared with the baseline BL (Fig. 5b), our method is more robust to illumination variations (first three input images), viewpoint changes (last two images) and appearance variations (heads in the 3rd image with white hats).

5 CONCLUSION

This paper proposes a novel crowd counting framework for zero-shot cross-domain crowd counting, which is closer to real-world counting applications. The proposed counting framework is built upon an external Momentum Template (MoT), which explicitly encodes the source domain information via the external representation. To effectively adapt to unseen target domains, we propose to learn to dynamically adapt the external MoT using an online target template extracted from the test image. With the dynamically adapted MoT, our model effectively finds dense head correspondences via the cross-correlation operation, thus providing a strong features for the final density map prediction. Extensive experiments on four large-scale crowd counting datasets demonstrate that our method achieves leading zero-shot cross-domain crowd counting performance without model fine-tuning, while also outperforming supervised domain adaptation methods that use fine-tuning on target domain data. Our method also obtains state-of-the-art performance on the source domain.

ACKNOWLEDGMENTS

This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11212518).

REFERENCES

[1] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, and P.H.S. Vedaldi. 2016. Fully-convolutional siamese networks for object tracking. In *ECCVW*. 850–865.

[2] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. 2018. Scale Aggregation Network for Accurate and Efficient Crowd Counting. In *European Conference on Computer Vision (ECCV)*. 734–750.

[3] Antoni B Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. 2008. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1–7.

[4] Antoni B. Chan and Nuno Vasconcelos. 2009. Bayesian Poisson regression for crowd counting. In *International Conference on Computer Vision*. 545–551.

[5] L. Chen, H. Zhang, and J. Xiao. 2017. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 5659–5667.

[6] J. Cheng, H. Xiong, and Z. Can. 2021. Decoupled Two-Stage Crowd Counting and Beyond. *IEEE Transactions on Image Processing* (2021), 2862–2875.

[7] Junyu Gao, Wei Lin, Bin Zhao, Dong Wang, Chenyu Gao, and Jun Wen. 2019. C^3 Framework: An Open-source PyTorch Code for Crowd Counting. *arXiv preprint arXiv:1907.02724* (2019).

[8] Junyu Gao, Qi Wang, et al. 2020. Feature-Aware Adaptation and Density Alignment for Crowd Counting in Video Surveillance. *IEEE Transactions on Cybernetics* (2020). https://doi.org/10.1109/TCYB.2020.3034316

[9] Weina Ge and R. Collins. 2009. Marked point processes for crowd counting. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2913–2920.

[10] Tao Han, Junyu Gao, Yuan Yuan, , and Qi Wang. 2020. Focus on Semantic Consistency for Cross-Domain Crowd Understanding. In *IEEE Conference on Acoustics, Speech and Signal Processing*. 1848–1852.

[11] Yutao Hu, Xiaolong Jiang, Xuhui Liu, Baochang Zhang, Jungong Han, Xianbin Cao, and David Doermann. 2020. NAS-Count: Counting-by-Density with Neural Architecture Search. *arXiv preprint arXiv:2003.00217* (2020).

[12] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. 2013. Multi-source Multi-scale Counting in Extremely Dense Crowd Images. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2547–2554.

[13] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. 2018. Composition Loss for Counting, Density Map Estimation and Localization in Dense Crowds. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 532–546.

[14] Xiaoheng Jiang, Li Zhang, Mingliang Xu, Tianzhu Zhang, Pei Lv, Bing Zhou, Xin Yang, and Yanwei Pang. 2020. Attention Scaling for Crowd Counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4706–4715.

[15] Di Kang and Antoni B. Chan. 2018. Crowd Counting by Adaptively Fusing Predictions from an Image Pyramid. In *British Machine Vision Conference*. 89.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Min Li, Zhaoxiang Zhang, Kaiqi Huang, and Tieniu Tan. 2008. Estimating the number of people in crowded scenes by MID based foreground segmentation and head-shoulder detection. In *International Conference on Pattern Recognition*. 1–4.

[18] Yuhong Li, Xiaofan Zhang, and Deming Chen. 2018. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1091–1100.

[19] Yuhong Li, Xiaofan Zhang, and Deming Chen. 2018. CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1091–1100.

[20] Liang Liu, Hao Lu, Hongwei Zou, Haipeng Xiong, Zhiguo Cao, and Chunhua Shen. 2020. Weighing Counts: Sequential Crowd Counting by Reinforcement Learning. *arXiv preprint arXiv:2007.08260* (2020).

[21] Lingbo Liu, Zhilin Qiu, Guanbin Li, Shufan Liu, Wanli Ouyang, and Liang Lin. 2019. Crowd counting with deep structured scale integration network. In *IEEE Conference on International Conference on Computer Vision*. 1774–1783.

[22] Lingbo Liu, Zhilin Qiu, Guanbin Li, Shufan Liu, Wanli Ouyang, and Liang Lin. 2019. Crowd counting with deep structured scale integration network. In *Proceedings of the IEEE International Conference on Computer Vision*. 1774–1783.

[23] Lingbo Liu, Hongjun Wang, Guanbin Li, Wanli Ouyang, and Liang Lin. 2018. Crowd Counting using Deep Recurrent Spatial-Aware Network. In *International Joint Conference on Artificial Intelligence*. 849–855.

[24] Wei Liu, Shengcai Liao, Weiqiang Ren, Weidong Hu, and Yinan Yu. 2019. Highlevel semantic feature detection: A new perspective for pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5187–5196.

[25] W. Liu, M. Salzmann, and P. Fua. 2019. Context-Aware Crowd Counting. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 5094–5103.

[26] Xiyang Liu, Jie Yang, and Wenrui Ding. 2020. Adaptive Mixture Regression Network with Local Counting Map for Crowd Counting. *arXiv preprint arXiv:2005.05776* (2020).

[27] Y. Liu, Z. Wang, and M. Shi. 2020. Towards Unsupervised Crowd Counting via Regression-Detection Bi-knowledge Transfer. In *ACM International Conference on Multimedia*. 129–137.

[28] J. Long, E. Shelhamer, and T. Darrell. 2017. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 4 (2017), 640–651.

[29] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. 2019. Bayesian loss for crowd count estimation with point supervision. In *IEEE International Conference on Computer Vision*. 6142–6151.

[30] Mark Marsden, Kevin McGuinness, Suzanne Little, Ciara E Keogh, and Noel E O'Connor. 2018. People, penguins and petri dishes: Adapting object counting models to new visual domains and object types without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8070–8079.

[31] Viresh Ranjan, Hieu Le, and Minh Hoai. 2018. Iterative Crowd Counting. In *European Conference on Computer Vision*. 278–293.

[32] Deepak Babu Sam and R. Venkatesh Babu. 2018. Top-Down Feedback for Crowd Counting Convolutional Neural Network. In *AAAI*.

[33] Deepak Babu Sam, Skand Vishwanath Peri, Mukuntha Narayanan Sundararaman, Amogh Kamath, and Venkatesh Babu Radhakrishnan. 2020. Locate, Size and Count: Accurately Resolving People in Dense Crowds via Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[34] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. 2017. Switching Convolutional Neural Network for Crowd Counting. In *IEEE Conference on Computer Vision and Pattern Recognition*. 4031–4039.

[35] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, and Ming-Ming Cheng. 2018. Crowd counting with deep correlation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5382–5390.

[36] Vishwanath A Sindagi and Vishal M Patel. 2019. Multi-level bottom-top and top-bottom feature fusion for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision*. 1002–1012.

[37] Vishwanath A Sindagi, Rajeev Yasarla, and Vishal M Patel. 2020. JHU-CROWD++: Large-Scale Crowd Counting Dataset and A Benchmark Method. *Technical Report* (2020).

[38] Jia Wan and Antoni B. Chan. 2019. Adaptive Density Map Generation for Crowd Counting. In *IEEE International Conference on Computer Vision*. 1130–1139.

[39] Jia Wan and Antoni B. Chan. 2020. Modeling Noisy Annotations for Crowd Counting. In *Advances in Neural Information Processing Systems*.

[40] Jia Wan, Ziquan Liu, and Antoni B. Chan. 2021. A Generalized Loss Function for Crowd Counting and Localization.. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[41] Jia Wan, Wenhan Luo, Baoyuan Wu, Antoni B. Chan, and W. Liu. 2019. Residual Regression With Semantic Prior for Crowd Counting. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 4031–4040.

[42] J. Wan, Qingzhong Wang, and Antoni B. Chan. 2020. Kernel-based Density Map Generation for Dense Object Counting. *IEEE transactions on pattern analysis and machine intelligence* PP (2020).

[43] Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai. to appear Dec 2020. Distribution Matching for Crowd Counting. In *Advances in Neural Information Processing Systems*.

[44] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. 2020. Nwpu-crowd: A large-scale benchmark for crowd counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[45] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. 2019. Learning from synthetic data for crowd counting in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*. 8198–8207.

[46] Q. Wang, Tao Han, Junyu Gao, and Yuan Yuan. 2021. Neuron Linear Transformation: Modeling the Domain Shift for Crowd Counting. *IEEE transactions on neural networks and learning systems* PP (2021).

[47] S. Woo, J. Park, and J.Y. Lee. 2018. Cbam: Convolutional block attention module. In *European Conference on Computer Vision (ECCV)*.

[48] Chenfeng Xu, Kai Qiu, Jianlong Fu, Song Bai, Yongchao Xu, and Xiang Bai. 2019. Learn to Scale: Generating Multipolar Normalized Density Maps for Crowd Counting. In *IEEE International Conference on Computer Vision*. 8382–8390.

[49] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. 2015. Cross-scene crowd counting via deep convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 833–841.

[50] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. 2016. Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In *IEEE Conference on Computer Vision and Pattern Recognition*. 589–597.

[51] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE Conference on International Conference on Computer Vision*. 2223–2232.