Neurocomputing 494 (2022) 418–431

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Discovering regression-detection bi-knowledge transfer for unsupervised cross-domain crowd counting

Yuting Liu [a], Zheng Wang [b], Miaojing Shi [c], Shin'ichi Satoh [d], Qijun Zhao [a,*], Hongyu Yang [a,*]

[a] College of Computer Science, Sichuan University, China
[b] School of Computer Science, Wuhan University, China
[c] King's College London, UK
[d] National Institute of Informatics, Japan

ARTICLE INFO

Article history:
Received 17 May 2021
Revised 3 January 2022
Accepted 23 April 2022
Available online 28 April 2022
Communicated by Zidong Wang

Keywords:
Unsupervised
Cross-domain crowd counting
Regression-detection bi-knowledge transfer
Co-training

ABSTRACT

Despite impressive progress in crowd counting over the last years, it is still an open challenge to reliably count crowds across visual domains. This paper addresses this setting, presenting an unsupervised cross-domain crowd counting framework able to perform unsupervised adaptation across domains with available unlabeled target data. We achieve this by learning to discover bi-knowledge transfer between regression- and detection-based models from a labeled source domain. The dual source knowledge of the two models is heterogeneous and complementary as they capture different modalities of crowd distribution. Specifically, we start by formulating the mutual transformations between the outputs of regression- and detection-based models as two scene-agnostic transformers which enable knowledge transfer between the two models. Given the regression- and detection-based models and their mutual transformers learnt on the source, we then introduce a self-supervised co-training scheme to encourage the knowledge transfer between the two models on the target. We further enhance the model adaptation with our modified mixup augmentation strategy. A thorough benchmark analysis against the most recent cross-domain crowd counting methods and detailed ablation studies show the advantage of our method.

1. Introduction

Crowd counting aims to estimate the number of persons in crowd images or videos. It has drawn much attention recently due to its important practical applications, such as public safety management, human behavior modeling, and smart city [1–5]. Recent years [6–8,3,9,10] have witnessed significant progress in the *closed set* crowd counting problem, where crowd counters are normally trained with limited crowd images and extensive instance-level annotations (points or boxes) for persons in crowd images. This however faces problems when it comes to the *open-set* problem in practice (*i.e.*unseen crowd and scenarios in new domain). On the one hand, directly applying the crowd counters trained on the existing observed domain (source) to the new domain (target) suffers from significant performance degradation owing to the domain-shift problem. On the other, annotating a large number of persons in unseen target images for re-training crowd counters would be extremely time-consuming and even unaffordable.

A few methods have been proposed recently to solve this challenging issue, namely the cross-domain crowd counting problem (CDCC). These methods can be broadly grouped into two major categories - cross-domain generalization (CDG) and cross-domain adaptation (CDA) - according to the availability of data in target domain during training. CDG methods attempt to enable cross-domain generalization towards target domain only with source data. For example, Shi et al. [11] learn an ensemble of decorrelated regressors to prevent model overfitting. Xu et al. [12] propose a Learning to Scale Module (L2SM) to enhance model robustness to density pattern shift. CDA methods address the problem in a setting where target domain data (*i.e.*unlabeled image) are also accessible during training. Their main focus is to study how to reduce the domain gap between source and target domains. These methods are mostly realized via the density regression-based models, where a density distribution is learnt for each crowd image whose integral over the density map gives the total count of crowd in that image. Some methods also employ detection-based models [5,13], where every individual is to be localized in the crowd images.

* Corresponding authors.
E-mail addresses: yuting.liu@stu.scu.edu.cn (Y. Liu), wangzwhu@whu.edu.cn (Z. Wang), miaojing.shi@kcl.ac.uk (M. Shi), satoh@nii.ac.jp (S. Satoh), qjzhao@scu.edu.cn (Q. Zhao), yanghongyu@scu.edu.cn (H. Yang).

https://doi.org/10.1016/j.neucom.2022.04.107
0925-2312/

The regression-based methods perform very well in high-density and congested crowds, as they do not predict individual locations. The detection-based methods, on the other hand, provide individual locations, and are believed to perform better in low-density and sparse crowds (see Fig. 1 (a)). We conduct a further comparison between the regression-based and detection-based methods in the cross-domain setting in Fig. 1 (b): two state-of-the-art methods [9,14] were trained on ShanghaiTech SHB [15] for density regression and head detection, respectively. With trained models, we directly predicted on ShanghaiTech SHA and draw the error distribution between the predicted counts and ground truth. The figure shows that two error distributions (denoted by RegNet and DetNet) are clearly separated; DetNet performs better (small errors) than RegNet in low-density areas while its performance significantly drops and underestimates the crowd count in high-density areas (green points in Fig. 1 (b)); in contrast, RegNet performs much better than DetNet in relatively high-density area (blue points in Fig. 1 (b)).

The observation above tells us that the pre-trained regression and detection models learnt on the source can compensate each other when deployed on the target, if we could combine their respective strengths in high- and low-density areas. This paper focuses on studying the cross-domain crowd counting problem in an unsupervised cross-domain adaptation setting, where the crowd counter is transferred from an annotated source domain to an unlabeled target domain. Unlike existing contributions, we propose to leverage knowledge from regression and detection-based models learnt on the source collaboratively, by letting different knowledge from heterogeneous models complement each other on the target. The main difficulty lies in how to construct, exchange and effectively combine the knowledge from regression and detection-based models.

We tackle this through modeling regression-detection bi-knowledge transfer on the source, and adapting the two models to the target by self-supervised co-training. The knowledge transfer can be modeled via constructing mutual transformations between the predictions of regression- and detection-based counting models. Transforming the detection results to combine with the regression is rather straightforward: each individual location can be convolved with a Gaussian kernel to generate the density distribution [15]. The density estimation result can be enhanced in this way especially in the low-density area [16]. On the other hand, transforming the density regression result to combine with the detection, though intuitive, was not explored before. Analog to convolution and deconvolution, we show the latter transformation could be formulated as the inverse operation of the former, where there exist different ways to single out a solution and we offer one in the deep-fashion (see Section 3.3). According to our formulation, the regression-detection mutual transformations are only dependent on the Gaussian kernel used when convolving at each individual location. As long as the Gaussian kernel is adopted in the same rule, the transformation between the regression and detection results can be regarded as two scene-agnostic transformers in crowd counting. They can be learnt from the source and used on the target.

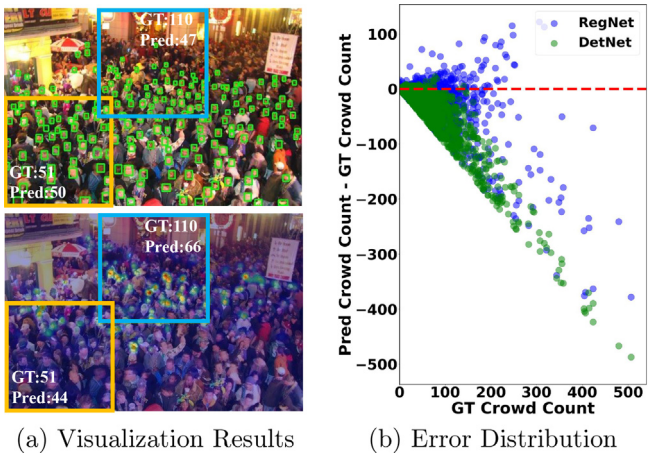


Fig. 1. (a) Comparison of visualization results by pre-trained DetNet (top) and RegNet (bottom) models when deployed on the unseen target image. 'GT' indicates the ground truth crowd counts, and 'Pred' indicates the predicted crowd counts. The yellow and blue squares mark the sparse crowds and dense crowds areas(b) The error crowd count (predicted crowd count - ground truth crowd count) *vs.* the ground truth count in image patches. The blue and green points show the error distribution by RegNet and DetNet, respectively. [Best viewed in color].

Given the regression- and detection-based models and their mutual transformers learnt on the source, we propose an iterative self-supervised co-training scheme to adapt the two models to the target: the two pre-trained regression- and detection-based models are first deployed on the target to obtain initial predictions from different perspectives; the regression and detection predictions pass through their mutual transformers to generate their counterparts, respectively. The initial and transformed regression (detection) predictions are fused to create the pseudo ground truth for density regression (individual detection). We use the two sets of pseudo ground truth to further fine-tune the regression and detection model, respectively. To make better use of the available data, we extend the mixup strategy [17] to perform data augmentation with labeled source and pseudo labeled target data during the co-training procedure, which are respectively implemented in image-level for detection and semantic-level for regression. The whole process repeats for several cycles until the convergence of the two models.

In summary, the main contributions of this work are as follows.

- We introduce a novel framework to learn to count and localize crowds on the unlabeled target domain via regression-detection bi-knowledge transfer modeling and self-supervised co-training. To the best of our knowledge, we are the first to discover the mutual knowledge transfer between regression- and detection-based models towards unsupervised cross-domain crowd counting.
- We investigate the mutual transformations between density regression and individual detection, and formulate them as two scene-agnostic transformers in the crowd counting.
- Thanks to the models and transformers learnt on the source, we propose a self-supervised co-training scheme on the target to fine-tune the regression and detection models with generated pseudo labels and boost the performance of both iteratively.
- We demonstrate the effectiveness of our method against other state-of-the-arts in the cross-domain setting with several standard benchmarks, *i.e.* ShanghaiTech [15], UCF_CC_50 [18] and UCF_QNRF [19].

This paper is an extension of our previous work [20] in the *ACM International Conference on Multimedia*(*Oral*). Compared with its preliminary conference version, this paper has been notably extended in several aspects.

- **Survey and comparison.** We make a more comprehensive survey on the cross-domain crowd counting task. Detailed descriptions and comparisons are presented in the "Introduction" and "Related Work".
- **Investigation on the transformations modeling.** We provide deeper investigations on solving the transformer from regression to detection, including newly added pseudo inverse solution.

- **Methodology.** Mixup strategy is explored to perform data augmentation for both regression and detection models during the self-supervised co-training procedure, which further boosts the performance compared to previous version [20].
- **Experiments and analysis.** Considerable new evaluation experiments and analyses are added for modeling transformer, generating samples for fine-tuning, and adapting models to the target.

The rest of this paper is organized as follows. Section 2 reviews related works about cross-domain crowd counting, regression- and detection-based crowd counting. Section 3 presents our method, including the architecture, base networks, regression-detection mutual transformations, and regression-detection bi-knowledge transfer. Section 4 reports experimental results, where we also conduct a batch of ablation studies to discuss how to model the transformer, how to generate samples, and how to adapt the model. Finally, conclusions are drawn in Section 5.

2. Related Works

2.1. Cross-domain crowd counting

Extending deep crowd counting models from a limited domain (source) to a new domain (target) is crucial for developing a scalable counting system. To better adapt the counting models to the target whilst minimizing annotation effort, a few methods have been proposed recently. One line of methods [15,12,11] focus on improving the cross-domain generalization ability of deep models only from source data. Typically, scale insensitive model architecture and loss function are designed. For example, Zhang et al. [15] address density scale shifts by proposing a multi-column network with different kernel sizes. Xu et al. [12] introduce center loss to condense the density scale distribution. Shi et al. [11] aim to learn an ensemble of correlation regularized regressors to prevent overfitting. As target information is not accessible, these methods usually achieve limited progress. Another line of methods [22,21,23,24] adopt cross-domain adaptation techniques by taking advantage of abundant whilst unlabeled data from the target. These methods generally seek to minimize the domain shifts on image-level or feature-level via adversarial training. For example, Wang et al. [22] establish a large-scale synthetic crowd dataset and introduced an SSIM Embedding (SE) Cycle GAN (based on [21]) to transform the synthetic image into photo-realistic style of the target domain. However, it faces the problem of the unsatisfying transformed image and requires manually selecting specific samples in the synthetic dataset. From the other side, [23,24] choose to align domain features in the semantic space. These methods employ coarse-grained alignment by aligning features globally and may not be able to align local content (*i.e.* context, density,*etc) mismatch between domains. More recently, wang et al. [25] propose to model the domain shift at the parameter-level and then transfer the source model to the target model through a Neuron Linear Transformation (NLT). Related to the second line of methods, our method is also formed in an unsupervised domain adaptation setting across real-world datasets; but we take a different method to reduce domain gaps by transferring regression-detection bi-knowledge with a self-supervised co-training scheme.*

2.2. Regression-based crowd counting

Regression-based methods [2,26,11,27–30] encode the spatial distribution of the crowd into a density map by convolving annotated head points with Gaussian kernels. They learn a mapping from the crowd image to the density map. The integral of the density map gives the crowd count in the image [31]. Researches in recent trends focus on designing more powerful DNN structures and exploiting more effective learning paradigms [32,4,33,9,6,10,34,35]. For example, Yuan et al. [36] propose a scale-communicative aggregation network to learn density map with high-resolution; Liu et al. [9] introduce an improved dilated multi-scale structure similarity (DMS-SSIM) loss to learn density maps with local consistency. Although regression-based methods have made remarkable progress in counting the number of persons in crowds, their performance on low-density crowds is not satisfying yet [16]. Besides, these methods are not capable of providing individual locations in the crowds, which, on the other hand, are believed to be the merits of detection-based crowd counting methods, as specified below.

2.3. Detection-based crowd counting

Detection-based methods detect precise locations of persons and estimate their counts via the number of detections. They are commonly adopted in relatively low-density crowds as the performance would decay severely in high-density crowds with small and occluded persons. A recent resurgence of detection-based methods in crowd counting [37,16,5,13] is owing to the advances of object detection in the deep learning context [38–41]. For example, Liu et al. [16] train an end-to-end people detector for crowded scenes depending on the annotations of person bounding boxes. Liu et al. [5] further design a weakly supervised detection framework by detecting persons only with point annotations. Despite the resurgence of detection-based methods, in terms of counting accuracy in dense crowds, they are still not as competitive as those regression-based methods, and often need to be integrated with the latter. Related methods [16,5,13] usually integrate regression- and detection-based models through an attention module in an implicit way [16,13]. These methods are all designed under the *fully-supervised closed set setting*. In this paper, we aim to transfer the knowledge between regression and detection-based models to co-train the two models on the target towards unsupervised cross-domain adaptation. The integration is performed explicitly by transforming the model predictions from one to another.

2.4. Comparison with previous works

We compare related state-of-the-art cross-domain crowd counting methods concerning the cross-domain setting, base network, and main focus. The summary is presented in Table 1. Our work has the following distinct features. 1) Our work considers combining the regression and detection models simultaneously. It is worthy of discovering a complementary relationship between two models and modeling it as scene-agnostic transformers. The transformers are learned from labeled source data and can be directly applied to the target. 2) Different from the *CDA* methods [21–24] that focus on adversarial domain adaptation, our method instead adopts a self-supervised co-training mechanism. We consider generating pseudo-ground truth for unlabelled target data as supervision to iteratively co-train existing regression and detection models. This is based on bi-knowledge transfer between the two models. 3) Different from the *CDG* methods [15,12,11,42] that focus on improving model generalizability with limited labeled source data, we consider making full use of rich unlabeled target data. To enhance model learning, an extended mixup strategy on counting problems is explored in our work, where we construct augmented training samples with labeled source and pseudo labeled target data.

Table 1
Summary of state-of-the-art cross-domain crowd counting methods. **Syn** and *Real* denote synthetic dataset and real dataset, respectively. Det (Reg) denotes the detection-(regression-) based model.

Group	Methods	Cross-domain setting Source→Target	Network	Focus
Cross-domain generalization (*CDG*)	MCNN [15]	*Real*→*Real*	Reg	multi-scale (image-based)
	L2S [12]	*Real*→*Real*	Reg	multi-scale (patch-based)
	D-ConvNet-v1 [11]	*Real*→*Real*	Reg	negative correlation learning
Cross-domain adaptation (*CDA*)	Cycle GAN [21]	**Syn**→*Real*	Reg	image translation
	SE Cycle GAN [22]	**Syn**→*Real*	Reg	image translation
	SE + FD [23]	**Syn**→*Real*	Reg	multi-task (crowd segmentation) + adversarial learning
	CODA [24]	*Real*→*Real*	Reg	multi-scale (patch-based) + adversarial learning
	Ours	*Real*→*Real*	Det + Reg	knowledge transfer + self-supervised co-training

3. Method

3.1. Problem, motivation and architecture

Suppose we have a labeled source crowd counting dataset $\mathscr{S}$ and an unlabeled target crowd counting dataset $\mathscr{T}$. Our task is to learn to count and localize persons in the $\mathscr{T}$ by adapting regression and detection models originally trained on the $\mathscr{S}$.

We find that, given a crowd image, the regression model performs better in high-density areas while the detection model is better in low-density areas. Above observations (in Fig. 1) clearly demonstrate the complementary effect between the regression and detection models. We can thus utilize the dual source knowledge of the two models from different perspectives to adapt them to the $\mathscr{T}$. To combine their strength, this dual source knowledge needs to be transformable between each other, and transferable from the source to the target. Transforming the detection result to the regression result is rather a standard procedure: using a Gaussian kernel to convolve at each detected individual location [15,16]. Its inverse problem, transforming the regression result to the detection result, however has not been exploited before. We show in Section 3.3 that there are several ways to single out a solution for this inverse transformation. Analog to deconvolution in deep learning, we offer a modern solution by modeling it with deep neural networks. The motivation of this paper is to collaboratively leverage the dual knowledge of regression- and detection-based models learnt on labeled $\mathscr{S}$ and encourage the two models to co-train each other on the unlabeled $\mathscr{T}$ to enhance their performance.

Fig. 2 presents an overview of our method. It consists of two parts, namely regression-detection mutual transformations modeling on the source $\mathscr{S}$ and regression-detection bi-knowledge transfer on the target $\mathscr{T}$. The former part models mutual transformations, *i.e.* the *Det-to-Reg* and *Reg-to-Det* transformers on the source. The latter part conducts bi-knowledge transfer between detection and regression models on the target with an iterative self-supervised co-training scheme.

In the following, we first briefly introduce the base networks we used and then formulate the regression-detection mutual transformations. For the *Reg-to-Det* transformer, we demonstrate two solutions. After that, we illustrate the bi-knowledge transfer, which consists of three key steps in each self-supervised co-training cycle. In particular, a mixup strategy is newly added to the training procedure.

3.2. Base networks

Before starting the technical details, we first introduce the regression and detection networks employed in this paper. They are learnt with ground truth annotations in the source dataset.

3.2.1. Regression network

We choose the deep structured scale integration network (DSSINet) [9] as our regression network *R*. It is good at addressing the scale shift problem by using conditional random fields (CRFs) for message passing among multi-scale features. DSSINet takes VGG16 [43] as its backbone and is trained on $\mathscr{S}$ with the proposed dilated multi-scale structural similarity loss. The output of the network is a crowd density map M^R.

3.2.2. Detection network

We adopt the center and scale prediction (CSP) based pedestrian detector [14] as our detection network *D*. CSP is an anchor-free keypoint-based detector that predicts the center point and scale of each pedestrian. CSP takes ResNet-50 [44] as its backbone and is trained with the cross-entropy classification loss and the smooth-L1 loss. The output of the network consists of an individual localization map M^D (0–1 map) and a scale map M^S indicating the person's location and size information, respectively.

3.3. Regression-detection mutual transformations

In this section, we formulate the mutual transformations between the output of the regression and detection models as *Det-to-Reg* Ψ and *Reg-to-Det* Φ, respectively.

3.3.1. Det-to-Reg Ψ

Det-to-Reg Ψ refers to the transformation from crowd density map to individual location map. This can be achieved in a rather standard way following [16,15]: we convolve at each nonzero point of the individual localization map M^D with a Gaussian kernel G_{σ_j} to produce the crowd density map $M^R(z)$,

$$M^R(z) = \Psi\left(M^D\right) = \sum_{j=1}^{H} \delta(z - z_j) * G_{\sigma_j}(z), \tag{1}$$

where z_j signifies the j-th nonzero pixel in M^D and H is the total number of nonzero pixels (heads) in M^D. σ_j is proportional to the person scale value at point j, *i.e.* $\sigma_j \propto M^S_j$. In practice, many crowd counting datasets only have head point annotations (M^D) available. When generating the ground truth density map, σ_j is either fixed (in sparse crowd scenes) or approximated via the distance d_j from person j to his/her K-nearest neighbors (in dense crowd scenes) [15], *i.e.* $\sigma_j = \beta d_j$. We adopt this as the default way to compute σ_j as it frees the usage of the person scale map M^S. More importantly, it is consistent with how we transform back from the crowd density to the individual localization and scale, as specified below.

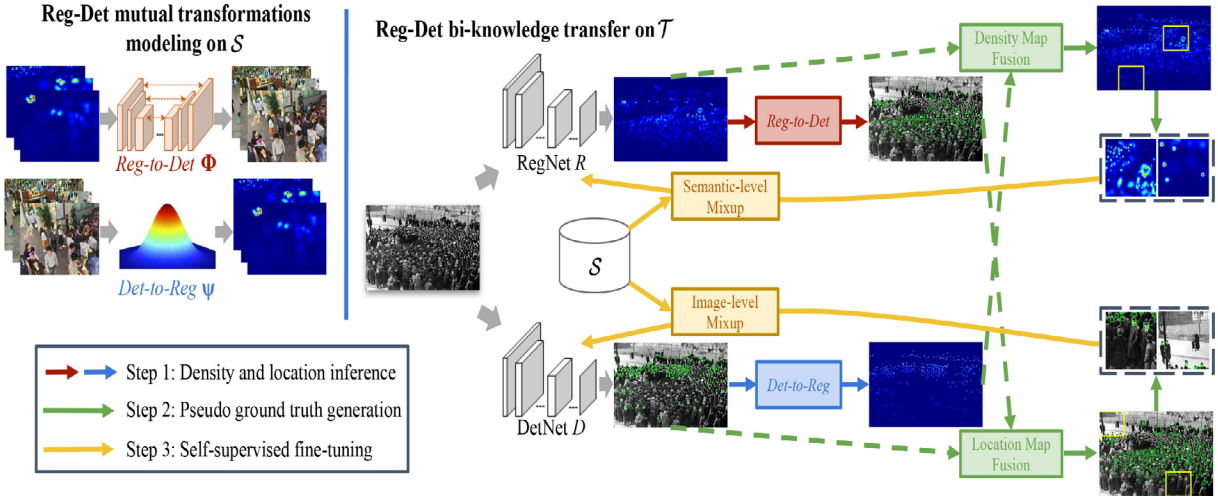


Fig. 2. Overview of our method: **Left**: We first model the *Det-to-Reg* Ψ and *Reg-to-Det* Φ transformers on the source dataset ($\mathcal{S}$). For the *Det-to-Reg* transformer, we model it as the gaussian transformation. For the *Reg-to-Det* transformer, we learn it by an encoder-decoder based model on the source dataset ($\mathcal{S}$). **Right**: we subsequently conduct regression-detection bi-knowledge transfer on the target ($\mathcal{T}$) via iterative self-supervised co-training. **In each iteration**: **Step1)** We feed the target image to the RegNet and DetNet to infer their initial predictions; **Step2)** We transform the predictions from one network to its counterpart and generate pseudo ground truth; **Step3)** We fine-tune the two models with reliable pseudo ground truth to facilitate them adapt to the target. Besides, We perform mixup data augmentation during training with labeled source and pseudo labeled target data, which is implemented on image-level for regression and semantic-level for detection respectively.

3.3.2. Reg-to-Det Φ

Reg-to-Det Φ models the transformation from the crowd density map M^R to the individual localization and scale maps (M^D, M^S). Based on the similar spirit above, M^S can be easily estimated by referring to the distance from each person detected in M^D to its nearest neighbors, or if we have another M^S, *e.g.* detection adapted from source to the target (see Section 3.4), the transformed scale map from M^R can be simply referred to values in the adapted scale map. Thus, the real target is to find the projection Φ from M^R to M^D,

$$M^D = \Phi\left(M^R\right). \tag{2}$$

This is an inverse operation to Eq. 1. Recovering M^D is indeed to find Φ to minimize the equation,

$$\Phi^* = \min_{\Phi} \|M^D - \Phi\left(M^R\right)\|_2 \tag{3}$$

In the following, we first offer a standard option to single out M^D, which simplifies the problem as solving the pseudo inverse. In light of the learnable convolution and deconvolution in deep learning, we then propose to learn a non-linear mapping with an encoder-decoder.

Pseudo inverse solution. Reg-to-Det transformer Φ indeed is the inverse transformation of the Det-to-Reg transformer Ψ. Recalling the definition of Ψ in Eq. 1, it is realized by convolving at every nonzero point (head center) of the individual location map M^D with a Gaussian kernel G_σ. Without loss of generality, we assume σ is fixed throughout density crowds, Eq. 1 can be rewritten as a matrix convolution between G_σ and M^D:

$$M^R = G_\sigma * M^D, \tag{4}$$

where G_σ is a $k \times k$ Gaussian kernel. Analog to the Compute Unified Device Architecture (CUDA) implementation of convolution in deep learning [45], it can be implemented via matrix multiplication: the $k \times k$ Gaussian kernel is unrolled as a $1 \times k^2$ vector $\widetilde{G_\sigma}$; similarly, we take $k \times k$ square at every pixel (center of the square) of M^D and unroll the square values as an $k^2 \times 1$ vector; for pixels near the edges of M^D, we pad the square with zeros. In this way, for N pixels in total in M^D, they form an $k^2 \times N$ matrix $\widetilde{M^D}$. Eq. 4 becomes

$$\widetilde{M^R} = \widetilde{G_\sigma} \cdot \widetilde{M^D}, \tag{5}$$

where $\widetilde{M^R}$ is of size $1 \times N$ and the unrolled version of M^R. $\widetilde{M^D}$ is directly related to M^D.

A standard approach to solve M^D relies on the inverse transform from Eq. 5. As $\widetilde{G_\sigma}$ is a non-singular matrix, we compute its pseudo inverse G_υ,

$$\widetilde{M^D} = \left(\widetilde{G_\sigma}^T \widetilde{G_\sigma}\right)^{-1} \widetilde{G_\sigma}^T \cdot \widetilde{M^R}. \tag{6}$$

where we obtain $G_\upsilon = \left(\widetilde{G_\sigma}^T \widetilde{G_\sigma}\right)^{-1} \widetilde{G_\sigma}^T$ as the Reg-to-Det transformer Φ.

Another traditional solution such as lasso regression can also be used to figure out a feasible solution for M^D. Albeit standard, the solution is not accurate, and the variance of the recovered values in $\widetilde{M^D}$ is large between the detected person locations and other locations. Moreover, the assumption of σ_j being fixed in Eq. 1 does not always hold, making the linear operations not always approachable. For these reasons, we offer non-linear learning of Φ with an encoder-decoder in the deep learning context.

Encoder-decoder solution. We employ the nested UNet [46] as an encoder-decoder with dilated VGG-16 structure [32] to learn the mapping from M^R to M^D. The output of the encoder-decoder $\Phi\left(M^R\right)$ is enforced to be as close as M^D with an MSE loss applied to every image:

$$L_{MSE} = \|M^D - \Phi\left(M^R\right)\|_2 = \frac{1}{N}\sum_{i=1}^{N}\left(M_i^D - \Phi\left(M_i^R\right)\right)^2 \tag{7}$$

where i signifies the i-th pixel in the map, and there are in total N pixels in the image.

Although a crowd in an image can be very dense, the localization of each individual is marked with only one pixel in M^D, meaning that M^D is rather sparse, $H \ll N$. To balance the loss contributions between non-zero and zero pixels in M^D, we are

motivated to adapt the focal loss [41], which specifically copes with the unbalance issue in object detection, into a focal MSE Loss in our scenario:

$$L_{Focal-MSE} = \frac{1}{N}\sum_{i=1}^{N}\kappa_i(1-p_i)^{\gamma}\left(M_i^D - \Phi\left(M_i^R\right)\right)^2, \tag{8}$$

where

$$p_i = \begin{cases} \text{sigmoid}\left(\Phi\left(M_i^R\right)\right) & \text{if} \quad M_i^D = 1 \\ 1 - \text{sigmoid}\left(\Phi\left(M_i^R\right)\right) & \text{otherwise} \end{cases} \tag{9}$$

and

$$\kappa_i = \begin{cases} 1 & \text{if} \quad M_i^D = 1 \\ 0.1 & \text{otherwise} \end{cases} \tag{10}$$

κ_i is a weighting factor that gives more weight on the nonzero pixels of M^D as its number is much less than that of the zero pixels. $(1-p_i)^{\gamma}$ is a modulating factor that reduces the loss contribution from easy pixels (*e.g.* $\Phi\left(M_i^R\right)$ with very large value at $M_i^D == 1$ or very small value at $M_i^D == 0$.) while focuses on those hard pixels. γ is a parameter ($\gamma = 2$ in practice) to smoothly adjust the rate for easy pixels to be gradually down-weighted.

We also adopt the Dilated Multiscale Structural Similarity loss in $L_{DMS-SSIM}$[1] to enforce the local patterns (mean, variance and covariance) of $\Phi\left(M^R\right)$ visually similar to M^D. Its parameter setting is the same with [9]. To this end, the final objective function for *Reg-to-Det* module Φ is

$$L_{\Phi} = L_{Focal-MSE} + L_{DMS-SSIM} \tag{11}$$

Both Ψ and Φ are scene-agnostic transformations in crowd counting. As long as the Gaussian kernel is designed with the same rule, we can use Ψ and Φ to exchange the knowledge between the regression and detection models on the target $\mathscr{T}$.

3.4. Regression-detection bi-knowledge transfer

In this session, we transfer the knowledge learnt from the labeled source dataset to the unlabeled target dataset. The transfer is bi-directional between the regression and detection models. It is realized in an iterative *self-supervised co-training* way initiated from the pre-trained regression and detection models, R_0 and D_0, in the source (see Section 3.2). Without loss of generality, we use R_t and D_t to denote the regression and detection model at t-th cycle. Three steps are carried out in each *self-supervised co-training* cycle: **1)** R_t and D_t are used to infer the crowd density and location maps for each image I in $\mathscr{T}$, respectively; **2)** the inferred maps from D_t (R_t) is fused with the transformed maps using $\Psi(\cdot)$ ($\Phi(\cdot)$) to generate pseudo ground truth; **3)** R_t and D_t are fine-tuned with the regression and detection pseudo ground truth to further update to R_{t+1} and D_{t+1}. We further enhance both model training by introducing a mixup augmentation strategy, including semantic-level mixup for regression and image-level mixup for detection. The whole process iterates for several rounds until the convergence of R and D. We describe each of these below.

3.4.1. Density and location inference

Given R_t and D_t, we use $R_t(I)$ and $D_t(I)$ to indicate the crowd density and individual localization maps per image I. They can be easily obtained by forwarding the network with R_t and D_t once, respectively.

3.4.2. Pseudo ground truth generation

Referring to Fig. 1, the dual prediction $R_t(I)$ and $D_t(I)$ on image I is a complement to each other. To take advantage of both, we propose to transfer the knowledge from one to another to further enhance the model, respectively. Using the mutual transformations discussed in Section 3.3, we can obtain the counterpart of $R_t(I)$ and $D_t(I)$ via $\Psi(D_t(I))$ and $\Phi(R_t(I))$. $R_t(I)$ is then fused with $\Psi(D_t(I)), D_t(I)$ is fused with $\Phi(R_t(I))$, correspondingly.

Regarding the fusion between $R_t(I)$ and $\Psi(D_t(I))$, we propose to use the detection confidence weight map W_t to act as a guidance. W_t is modified such that within each $k \times k$ (same k for the Gaussian kernel) area of a detection center, the weights are set the same as the center weight. The fused regression result M^{R_t} is given by,

$$M^{R_t} = (1 - W_t) \cdot R_t(I) + W_t \cdot \Psi(D_t(I)). \tag{12}$$

The reason to use W_t is that the detector D_t normally performs better in low-density area with high confidence scores; thus its transformed regression result $\Psi(D_t(I))$ contributes more in the low-density area of M^{R_t} if multiplying it by W_t; the original $R_t(I)$ instead contributes more in the high-density area in Eq. 12.

Regarding the fusion between $D_t(I)$ with $\Phi(R_t(I))$, the spirit is similar: the transformed detection result $\Phi(R_t(I))$ from $R_t(I)$ produces more detections than $D_t(I)$ in the high-density area, while $D_t(I)$ normally produces less detections compared to the ground truth (see Fig. 1 (b)). We can simply fuse the detection from $D_t(I)$ with $\Phi(R_t(I))$ followed up by non-maximum suppression (NMS) [47], which should result in adequate detection in both high- and low-density area. We denote by M^{D_t} the final detection result. Notice that $\Phi(R_t(I))$ only produces individual center locations but not scales (sizes). In order to restore complete bounding boxes, we find the corresponding scales M^{S_t} using the original scale map M^S from D_t in the lower half of the image and nearest neighbor distances in the upper half of the image.[2]

Having received M^{R_t} and M^{D_t} (M^{S_t}), we select two patches of size 224×224 from each image[3], their pseudo labels are cropped correspondingly from the map. For M^{R_t}, we traverse all the non-overlapped patches with their densities and select the ones whose feature embedding is most similar to the source distribution. For M^{D_t}, we also traverse all the non-overlapped patches and find the ones with average detection confidence scores being the highest.

Algorithm 1: Training flow on the target $\mathscr{T}$.

Inputs: Target dataset $\mathscr{T}$, RegNet R, DetNet D, *Det-to-Reg* Ψ, *Reg-to-Det* Φ, Total iterations T
Outputs: R_T and D_T
$t = 0$
while $t < T$ **do**
  for $I \in \mathscr{T}$ **do**
    // Step 1: Density and location inference
    Infer $R_t(I)$ with R_t
    Infer $D_t(I)$ with D_t
    // Step 2: Pseudo ground truth generation
    // **Regression-detection knowledge transfer
    Transform $R_t(I)$ into $\Phi(R_t(I))$ with Eq. 1
    Transform $D_t(I)$ into $\Psi(D_t(I))$ with Eq. 2
    // **Regression-detection combination
    Fuse $R_t(I)$ with $\Psi(D_t(I))$ to obtain M^{R_t} with Eq. 12
    Fuse $D_t(I)$ with $\Phi(R_t(I))$ to obtain M^{D_t} with NMS

(*continued on next page*)

[1] Refer to [9] for implementation details.

[2] Crowds in the upper half of the image are usually dense, we observe that the nearest neighbor distances are closer to the real head scales than scale map M^S.

[3] The setting up is fixed for all datasets.

a (*continued*)

Algorithm 1: Training flow on the target $\mathscr{T}$.

```
// **Sampling training data
Select patch samples from M^{R_t} and M^{D_t} as U_tgt
end for
// Step 3: Fine-tuning
Fine-tune R_t on U_tgt to update to R_t + 1
Fine-tune D_t on U_tgt to update to D_t + 1
t = t + 1
end while
Return R_T and D_T
```

3.4.3. Self-supervised fine-tuning

We fine-tune R_t and D_t with the samples selected from every image alongside their pseudo ground truth obtained above. The updated models are denoted by R_{t+1} and D_{t+1}. Having the updated model, we could repeat the whole process to re-select samples and re-train the two models until their convergence at the T-th iteration, and get R_T and D_T. Algorithm 1 provides an overview of the training flow.

3.4.4. Data augmentation via mixup

To further enhancing the training robustness of both regression and detection models on the target, we introduce data augmentation via mixup [17], including semantic-level mixup for regression and image-level mixup for detection, specifically. Below we first briefly introduce the concept of mixup and then detail its implementing details for regression and detection.

In a nutshell, mixup encourages the model to behave linearly in-between training samples. It enforces the model predictions of mixed training sample pairs consistent with their mixed labels. We formally define the mixup operation as follows:

$$\lambda \sim Beta(\alpha, \alpha), \qquad Mix_\lambda(a,b) = \lambda \cdot a + (1-\lambda)\cdot b \tag{13}$$

The mixup-generated data $Mix_\lambda(a,b)$ are linearly interpolated from existing sample pairs (a,b) according to a random coefficient λ. α is the hyper-parameter in $Beta(\cdot)$ distribution that controls the strength of interpolation.

Semantic-level mixup for regression. Few works [48] explored how to apply mixup for crowd counting. Recently Zhao et al. [48] introduce mixup to encourage distribution alignment, where the mixed representations are indistinguishable for the distribution classifier. While, it still designs based on the classification paradigm. Unlike the traditional usage of mixup for classification via interpolating image pixels/ representations and one-hot labels [17,49], we consider extract representations of high-level semantics [50,51] (*e.g.* crowd densities) from deep layer of the regression network to make density map predictions from mixed representations of crowds consistent with the mixed density map labels, and thus perform semantic-level mixup[4]. We empirically find this strategy shows its effectiveness for training regression model, especially when performed with pseudo-labeled target data (see Section 4.6). As deeper feature representations encode valid information related to crowd density, we believe interpolations in-between representations of different density-level crowd data can enrich density distributions in semantic feature space. Meanwhile, the predictions of mixup generated semantic features naturally are consistent with their mixed density maps (analog to word embeddings, *e.g.* congested + sparse $\simeq$ Mid-level). Formally, given a pair of input images with their corresponding density map labels $(x1, y1), (x2, y2)$, we force predictions of mixed feature representations of $x1$ and $x2$ equal to their mixed density maps as follows:

$$f(Mix_\lambda(g(x1), g(x2))) = Mix_\lambda(y1, y2), \tag{14}$$

where $g(\cdot)$ denotes the mapping from input image data to the learnt semantic feature representations and $f(\cdot)$ denotes the mapping from the semantic representations to the outputs.

Image-level mixup for detection. For the detection task, we directly mixup image pixels between pairs of training samples to preserve the spatial property of objects. To avoid image distortion, we maintain the original geometry of image pairs (without resize or crop operation). The size of the mixed image is decided by the largest side of the image pair. At the same time, object labels of previous image pairs are directly merged (refers to [52]). The image-level mixup operation here can be viewed as a part of image pre-processing techniques, *e.g.* cropping, random occlusion, cut-and-paste.

4. Experiments

4.1. Datasets

ShanghaiTech [15]. It consists of 1,198 annotated images with a total of 330,165 people with head center annotations. This dataset contains two parts: SHA and SHB. The crowd images are sparser in SHB compared to SHA: the average crowd counts are 123.6 and 501.4, respectively. We use the same protocol as [15] that 300 images for training and 182 images for testing in SHA; 400 images for training and 316 images for testing in SHB.

UCF_CC_50 [18]. It has 50 images with 63,974 head center annotations in total. The headcount range between 94 and 4,543 per image. The small dataset size and large variance make it a very challenging counting dataset. Following [18], we perform 5-fold cross-validation to report the average test performance.

UCF_QNRF [19]. It is a large crowd counting dataset with 1535 high-resolution images and 1.25 million head annotations, among which 334 images are used as the testing set. The dataset contains extremely congested scenes where the maximum count of an image can reach 12865.

4.2. Implementation details and evaluation protocol

4.2.1. Implementation details

Base networks training. Training details of the DSSINET for regression and CSP for detection on the source follow the same protocol with [9,14], as specified in Section 3.2. Notice that the source crowd counting datasets do not provide bounding box annotations for training CSP, we thus train it on the source with point annotations following [5].

Reg-to-Det Φ. To solve the pseudo-inverse for Φ, we set $\sigma = 4$ and $k = 15$ in Eq. 4, following the default setting of the fixed Gaussian kernel [31]. To train the encoder-decoder for Φ, we randomly crop 224×224 patches from the ground truth density maps in $\mathscr{S}$. We initialize the first 10 convolutional layers of encoder-decoder with the weights from a VGG16 [43] network pre-trained on the ILSVRC classification task [53]. The rest convolutional layers are initialized via a Gaussian distribution with zero mean and standard deviation of 1×10^{-2}. The encoder-decoder is optimized by Adam with a learning rate of 1×10^{-5}. When testing, the output matrix of the encoder-decoder is binarized with a threshold of 0.2. We further merged redundant non-zero pixels within the local area (10×10) to obtain the final binary matrix.

[4] Here we extract the features from the last block (conv4_3) of the RegNet backbone (VGG16 based).

Self-supervised Co-training. For co-training the regression and detection models in the target set $\mathcal{T}$, Adam optimizer is used with a learning rate of 1×10^{-6} and 1×10^{-5} respectively. For performing mixup, a beta distribution with a parameter 2.0 is used in all experiments.

Cross-domain testing. We get the crowd counting and detection results by merging the outputs of R_T and D_T with the same procedure as in training.

4.2.2. Evaluation protocol

Counting performance. To measure the counting performance, we adopt the commonly used mean absolute error (MAE) and mean square error (MSE) [26,54] to compute the difference between the counts of ground truth and estimation.

Localization performance. To report localization performance, we measure the mAP for person head localization. Those predicted head points within a particular distance of c pixels to its nearest ground truth point are regarded as *true positives*. For a certain ground truth point, if there exist duplicate predictions that satisfy the condition, we choose the one with the highest score as true positive. Others are taken as *false positives*. Average precision (AP) is computed for every c and the final mAP is obtained as the average value of AP with various c. c is varied from 1 to 100 similar to [19].

4.3. Comparisons with state-of-the-arts

To show the effectiveness of our method, we compare against other state-of-the-arts [21–23,15,11,42,12] in cross-domain setting. Methods such as MCNN [15], D-ConvNet-v1 [11], and SPN + L2SM [12] learn the model from a real source dataset like ours. Note that these methods only perform the cross-domain evaluation to show their generalization ability on target datasets. Methods including Cycle GAN [21], SE Cycle GAN [22], and SE + FD [23] transfer the knowledge from a very large-scale synthetic dataset GCC [22], which contains 15,212 high-resolution images. Same as our problem setting, Method CODA [24] learns the model using labeled real source and unlabeled target datasets.

We present the results of cross-domain transferring from SHA (A) to SHB (B), UCF_CC_50 (C), and UCF_QNRF (Q), as well as from SHB to SHA and UCF_QNRF in Table 2. When transferring between SHA and SHB, our method performs the lowest MAE, *i.e.* 11.6 for A → B, and 103.6 for B → A, which improves other state of the arts with a big margin. When transferring in a more difficult setting, *i.e.* from SHA/ SHB to the large-scale dataset UCF_QNRF, our method produces significantly better results on both A → Q and B → Q over others; Compared with [21–23], which learn from the much larger and more diverse synthetic source GCC, our method shows very satisfying transferability. The result of A → C is relatively inferior to the best in prior arts. This is caused by the inferior detection result as crowd scenes in UCF_CC_50 are too congested to obtain satisfying localization results. Our method also competes [24], which adapts across real datasets in an unsupervised manner by utilizing adversarial learning and surrogate crowd ranking task.

We also provide results compared with our baseline approaches denoted by RegNet and DetNet. They use either the regression network [9] or the detection network [14] trained from the source to directly predict the crowd density or individual localization in the target. As shown in Table 2, compared to our method, they are substantially inferior in terms of both counting and detection accuracy. This, on the other hand, illustrates the effectiveness of our method combining the strength of both models and delivering much more competitive results. We provide results of fully training on target data where we use RDBT (*UB*) to denote. These can be viewed as the upper-bounds of cross-domain training. Note that target data only provide head point annotations, we follow [9,5] to train the regression and the detection network on each target train set and evaluate the MAE, MSE and AP scores on according target test set. We also conduct the experiments taking the same backbone (VGG16) for both regression and detection networks (see Table 2: RDBT (*VGG16*)). The overall counting and detection results are lower than that with different backbones (see Table 2: RDBT). While the main contribution of this work is to propose a framework to leverage knowledge from regression and detection networks. The counting performance can be improved with better detection and regression backbones Comparing our augmented approach using mixup design (RDBT *w/ MAD*) to previous effort (RDBT), clear performance gains are observed on all target datasets. This verifies the effectiveness of our mixup design with pseudo-labeled target data. We provide more analysis in Table 7 for better understanding its mechanism.

Overall, our method achieves the best counting accuracy in most of the cross-domain settings. More importantly, we would like to point out that our method is also capable of providing precise individual localization of the crowds (see mAP in Table 2), which is another advantage over the state-of-the-arts.

4.4. Discussion on learning Reg-to-Det Φ

In this session, we conduct a series of studies to analyze the quality of learned Φ provided by pseudo inverse and encoder-decoder on various affecting factors (Table 3 & 4 & 5). We evaluate the quality of Φ by measuring the detection mAP of the trans-

Table 2
Comparisons with the state-of-the-art methods in the cross-domain setting from **Source**→**Target**. ^{*syn} indicate methods are instead learning from the synthetic dataset GCC.

	Methods	A→B			A→C			A→Q			B→A			B→Q		
		MAE↓	MSE↓	mAP↑	MAE↓	MSE↓	mAP↑	MAE↓	MSE↓	mAP↑	MAE↓	MSE↓	mAP↑	MAE↓	MSE↓	mAP↑
CDG	MCNN [15]	85.2	142.3	–	397.7	624.1	–	–	–	–	221.4	357.8	–	–	–	–
	D-ConvNet-v1 [11]	49.1	99.2	–	364.0	545.8	–	–	–	–	140.4	226.1	–	–	–	–
	SPN+L2SM [12]	21.2	38.7	–	332.4	425.0	–	227.2	405.2	–	126.8	203.9	–	–	–	–
CDA	Cycle GAN^{*syn} [21]	25.4	39.7	–	404.6	548.2	–	257.3	400.6	–	143.3	204.3	–	257.3	400.6	–
	SE Cycle GAN^{*syn} [22]	19.9	28.3	–	373.4	528.8	–	230.4	384.5	–	123.4	193.4	–	230.4	384.5	–
	SE+FD^{*syn} [23]	16.9	24.7	–	–	–	–	221.2	390.2	–	129.3	187.6	–	221.2	390.2	–
	CODA [24]	15.9	26.9	–	–	–	–	–	–	–	–	–	–	–	–	–
	RegNet [9]	21.6	37.5	–	419.5	588.9	–	198.7	329.4	–	148.9	273.8	–	267.2	477.6	–
	DetNet [14]	55.4	90.0	0.571	703.7	941.4	0.258	411.7	731.3	0.404	242.7	400.8	0.489	411.7	731.3	0.404
	RDBT [20]	13.3	29.2	0.757	368.0	518.9	0.518	175.0	294.7	0.546	112.2	218.1	0.661	211.3	381.9	0.535
	RDBT (*VGG16*)	15.6	30.1	0.728	374.6	526.4	0.469	189.8	310.9	0.496	124.8	229.6	0.628	222.8	396.1	0.483
	RDBT*w/ MAD* (Ours)	**11.6**	**21.0**	**0.770**	361.3	504.5	**0.537**	**172.8**	**291.9**	**0.557**	**103.6**	200.8	**0.689**	**205.8**	**380.2**	**0.546**
	RDBT (*UB*)	6.8	10.3	0.865	216.9	302.4	0.635	99.1	159.2	0.673	60.6	96.0	0.813	99.1	159.2	0.673

Table 3
Ablate various s & c of pseudo inverse solution in the setting A → Q & A → B. mAP is reported for individual localization performance.

Solution	s	c	A→Q	A→B
Pseudo inverse	0.001	10	0.120	0.197
	0.001	20	0.182	0.294
	0.001	30	0.189	0.304
	0.001	40	0.169	0.277
	0.0005	30	0.201	0.321
	0.00025	30	0.229	0.363
	0.000125	30	0.218	0.344
Encoder-Decoder	0.2	10	**0.448**	**0.613**

formed individual localization maps at iteration 0 and report results in the setting of A → Q & A → B. We also visualize the quality of Φ in Fig. 3 for better comparison.

4.4.1. Analysis on pseudo inverse

As stated in Section 3.3.2, pseudo inverse offers a standard option to solve the *Reg-to-Det* transformer. For this part, We provide the results of pseudo inverse under various setups and compare it to the encoder-decoder in deep learning. Recall that the recovered $\widetilde{M^D}$ (M^D) by pseudo inverse is not sparse enough. To infer the individual locations, we need to first binarize the output matrix with a threshold s such that person locations are flagged with the 1-values in the matrix. Next, to remove redundant detections, we need another parameter c where 1-values within $c \times c$ area are indeed merged to represent one person.

Table 3 offers the results of pseudo inverse with different s and c. It can be seen that by enlarging c from 10 to 30 (pixels), the mAP is indeed improved, *e.g.* from 0.197 ($s = 0.001, c = 10$) to 0.304 ($s = 0.001, c = 30$) on SHB. By further lowering s from 0.001 to 0.00025, the mAP reaches 0.363. The performance drops if we keep increasing c or decreasing s. This suggests that the recovered matrix from pseudo inverse is very noisy and thus careful calibration of parameters is needed. Nevertheless, the results of pseudo inverse are even much lower than the encoder-decoder solution, where we got mAP 0.448 and 0.613 on QNCF and SHB, respectively. This verifies our statement that solutions such as pseudo inverse are not accurate and can only be taken as baseline.

To better visualize the obtained results, we illustrate some examples in Fig. 3. It shows that our proposed encoder-decoder provides a better solution for modeling Reg-to-Det Φ and the learned transformer can generalize well on unseen data.

4.4.2. Analysis on encoder-decoder

We apply several loss designs to learn encoder-decoder and report qualitative results in Table 4 to compare different loss designs. It shows that the proposed focal MSE loss $L_{Focal-MSE}$ demonstrates a strong superiority over the conventional MSE loss L_{MSE} (*e.g.* 0.347 *vs.* 0.423 on A → Q). As predicting the detection map can also be viewed as the traditional binary classification problem, we also compare the focal MSE loss with traditional focal cross-entropy [41] loss $L_{Focal-CE}$. We find that focal MSE loss is more suitable in our case than the focal loss (*e.g.* 0.423 *vs.* 0.220 on A → Q). Besides, adding the DMS-SSIM loss [9] further improves the result (L_Φ) to the best (*e.g.* 0.448 on A → Q). This justifies the usage of focal MSE loss and DMS-SSIM loss properly.

Another interesting thing worth mentioning is that, it seems to be sufficient to learn the encoder-decoder (Φ) with limited data in the source. We can see from Table 5 that with 30% data, the mAP for A → Q and A → B has already reached 0.406 and 0.610, *vs.* 0.448 and 0.613 with 100% data. We believe that this is another evidence to prove the scene-agnostic property of *Reg-to-Det* transformer: learning it from a small amount of data should be sufficient enough to achieve a reliable solution generalized over a large amount of data.

4.5. Discussion on training sample generation

As introduced in Section 3.4, we recursively feed the regression and detection networks with reliable pseudo labeled samples to adapt the two models to the target. The training sample generation pipeline is mainly composed of pseudo ground truth fusion and patch sampling. To study the contribution of the above components to the final cross-domain counting performance, we design

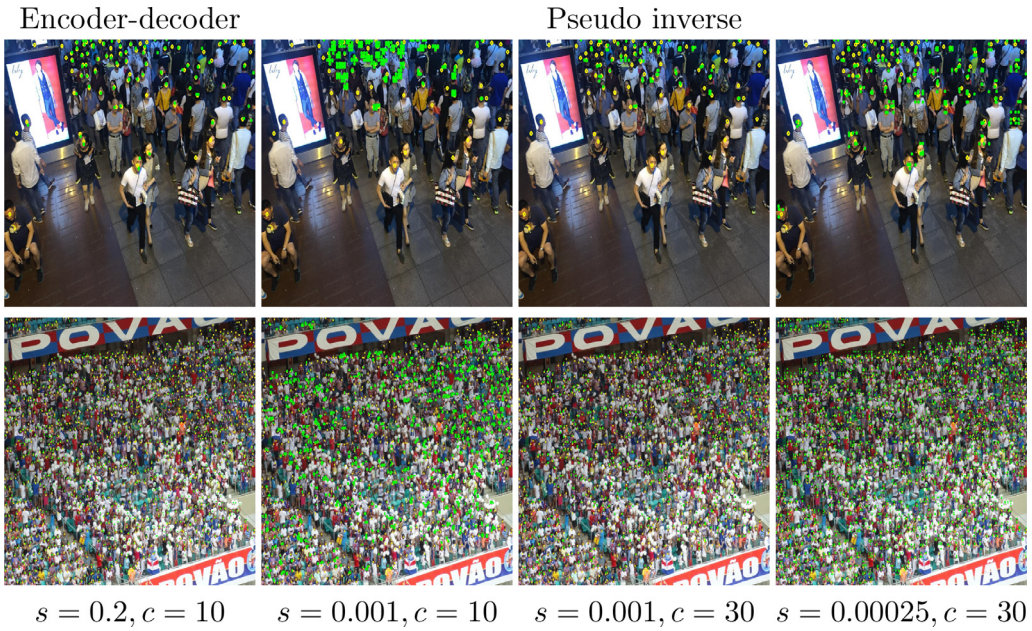


Fig. 3. Localization results provided by encoder-decoder and pseudo inverse on SHB (top) & UCF_QNRF (bottom) datasets with various s and c. Ground truth locations are marked with yellow dots while the predicted locations are in green.

Table 4
Ablate various loss designs of encoder-decoder solution in the setting of A → Q & A → B. mAP is reported for individual localization performance.

Solution	Loss design	A→Q	A→B
Encoder-decoder	L_{MSE}	0.347	0.482
	$L_{Focal-MSE}$	0.423	0.599
	$L_{Focal-CE}$	0.220	0.285
	L_{Φ}	**0.448**	**0.613**

Table 5
Ablate various data scales for training encoder-decoder in the setting of A → Q & A → B. mAP is reported for individual localization performance. `percentage%` A (`#samples`) indicates the amount of training samples.

Solution	Data scale	A→Q	A→B
Encoder-decoder	5% A (65)	0.295	0.422
	10% A (130)	0.323	0.463
	15% A (195)	0.402	0.603
	20% A (261)	0.386	0.572
	25% A (326)	0.378	0.567
	30% A (391)	0.406	0.610
	50% A (652)	0.373	0.546
	80% A (1044)	0.436	0.602
	100% A (1305)	0.448	0.613

careful ablation experiments as shown in Table 6. First, the output of R_t and D_t are not fused via the proposed regression-detection transformers (*e.g.* Eq. 12) but instead they are fine-tuned with their own pseudo ground truth. Patch sampling within the pseudo ground truth maps follows the same procedure as in Section 3.4 to choose the reliable and discriminative patches per image. The training iterates several cycles until the convergence of R_T and D_T. We present their results separately with the notation RDBT *w/o fusion*. For instance, for the regression result, the MAE and MSE are 146.7 and 275.3, respectively; by using the fusion, they can be reduced to 112.2 (-34.5) and 218.1 (-57.2). Similarly, for the detection result, the MAE and MSE significantly decrease 127.6 and 184.6 points, respectively; while the AP increases 19.4%. This demonstrates the effectiveness of knowledge transfer between the regression and detection networks.

Another detail of the training sample generation lies in the patch sampling strategy, where we propose to sample patches from the area similar to the source dataset for the regression fine-tuning; from the high confidence area for the detection fine-tuning. To justify this strategy, we compare it with random sampling in Table 6. It can be seen that our results are significantly better than random sampling (denoted by RDBT *w/ RS*) for both regression and detection results. Notice that the output of the regression and detection networks in RDBT *w/ RS* are fused in the same way as with ours.

Table 6
We show ablations of our pseudo ground truth generation. *w/o Fusion* means regression and detection outputs are not fused; *w/ RS* means training patches are randomly selected from the fused density or localization maps. We report final counting and individual localization results (MAE/MSE/mAP) in the setting of B → A.

B→A	Regression		Detection		
	MAE↓	MSE↓	MAE↓	MSE↓	mAP↑
RDBT *w/o Fusion*	146.7	275.3	251.7	407.0	0.467
RDBT *w/ RS*	152.1	281.0	165.8	302.9	0.609
RDBT	**112.2**	**218.1**	**124.1**	**222.4**	**0.661**

4.6. Discussion on mixup augmentation

To better apply mixup to our domain adaptive crowd counting task, we first investigate its several variants on training regression model: Mixup on target domain (*MT*), Image-level mixup (*IM*), Mixup on source domain (*MS*) as well as Mixup across domains (*MAD*). 1) *MT*: We train the regression model using mixup with pseudo-labeled target samples and their semantic-level mixed samples. 2) *IM*: Similar to *MT*, we while implementing the mixup combination using raw input images instead of feature representations of sample pairs. 3) *MS*: We train the regression model with labeled source samples, pseudo-labeled target samples, and semantic-level mixed samples on the source domain. 4) *MAD*: We train the regression model with labeled source samples, pseudo-labeled target samples, and semantic-level mixed samples across the source and target domains. Note that for *MT* and *IM*, we use a single data loader to obtain one mini-batch from the target, and mixup is then applied to the same mini-batch after random shuffling (refer to [17]). The obtained target and mixed data are equally sized. For *MS* and *MAD*, two mini-batches samples are obtained from the source and target respectively with different data loaders.

Three interesting observations can be made from Table 7. 1) Compared with our previous training scheme (RDBT), *MT* is helpful to enhance performance on the counting task. As we observe an oscillation phenomenon when training RegNet before, we think mixup helps to alleviate this problem. 2) We find that *IM* does not benefit model training, which also shows the significance of performing semantic-level mixup for regression. 3) As we can also access labeled samples from the source dataset, we thus operate mixup with labeled source and the pseudo-labeled target samples (*MAD*). Compared with *MT* and *MS*, we find that *MAD* has better counting performance and is thus chosen as our final design.

4.7. Discussion on self-supervised fine-tuning

Table 8 illustrates the results of the regression and detection models along with the increase of iterations. Both of the regression and detection models benefit from the iterative fine-tuning as the

Table 7
We show ablations of different mixup design: *w/ MT*, *w/ IM*, *w/ MS*, *w/ MAD*. Final counting and individual localization results (MAE/MSE/mAP) are reported in the setting of B → A.

B→A	Regression		Detection		
	MAE↓	MSE↓	MAE↓	MSE↓	mAP↑
RDBT [20]	112.2	218.1	124.1	222.4	0.661
RDBT w/ MT	107.2	210.3	123.0	214.5	0.665
RDBT *w/ IM*	118.7	226.1	127.4	231.9	0.656
RDBT *w/ MS*	139.5	251.8	130.5	237.5	0.652
RDBT *w/ MAD* (Ours)	**103.6**	**200.8**	**112.1**	**210.6**	**0.689**

Table 8
We show results (MAE/MSE/mAP) for different number of iterations in the setting of B → A.

B→A	Regression		Detection		
	MAE↓	MSE↓	MAE↓	MSE↓	mAP↑
Iteration 0	148.9	273.8	242.7	400.8	0.489
Iteration 1	130.3	245.6	165.7	300.8	0.610
Iteration 2	117.7	219.5	130.3	246.9	0.630
Iteration 3	109.3	210.5	120.1	222.2	0.669
Iteration 4	103.6	200.8	112.1	210.6	0.689
Iteration 5	105.2	204.0	114.7	212.1	0.688

iteration increases. The performance gets stable after several iterations when the improvement becomes marginal on both regression and detection. Normally, we stop when the mean absolute difference (MAE) of two consecutive iterations smaller than a thresh. In Fig. 4, we give an example of visualization results of our method in different iterations. It demonstrates that the results of DetNet and RegNet become better as adapting to the target dataset with more iterations. In Fig. 5, We provided the error distribution before (Iteration 0) and after (Iteration 4) the self-supervised fine-tuning to make a comparison. The figure shows that the overall counting error of our Reg-Net and Det-Net decreased; The Reg-Net performs better on low-density areas (see blue dots in Fig. 5 (b)) while the Det-Net performs better on relative high-density areas (see green dots in Fig. 5 (b)). The results demonstrate the effectiveness of the regression-detection bi-knowledge transfer.

4.8. Visualization results across unseen scenes

We show additional visualization results in some unseen scenes (from Google Images) in Fig. 6 and compare the obtained results with previous state-of-the-art approaches such as CSRNET [32] and DSSINET [9] learnt on SHB dataset. Obviously, our method has better visualization results, which proves that our method can reliably adapt across real-world unseen scenes.

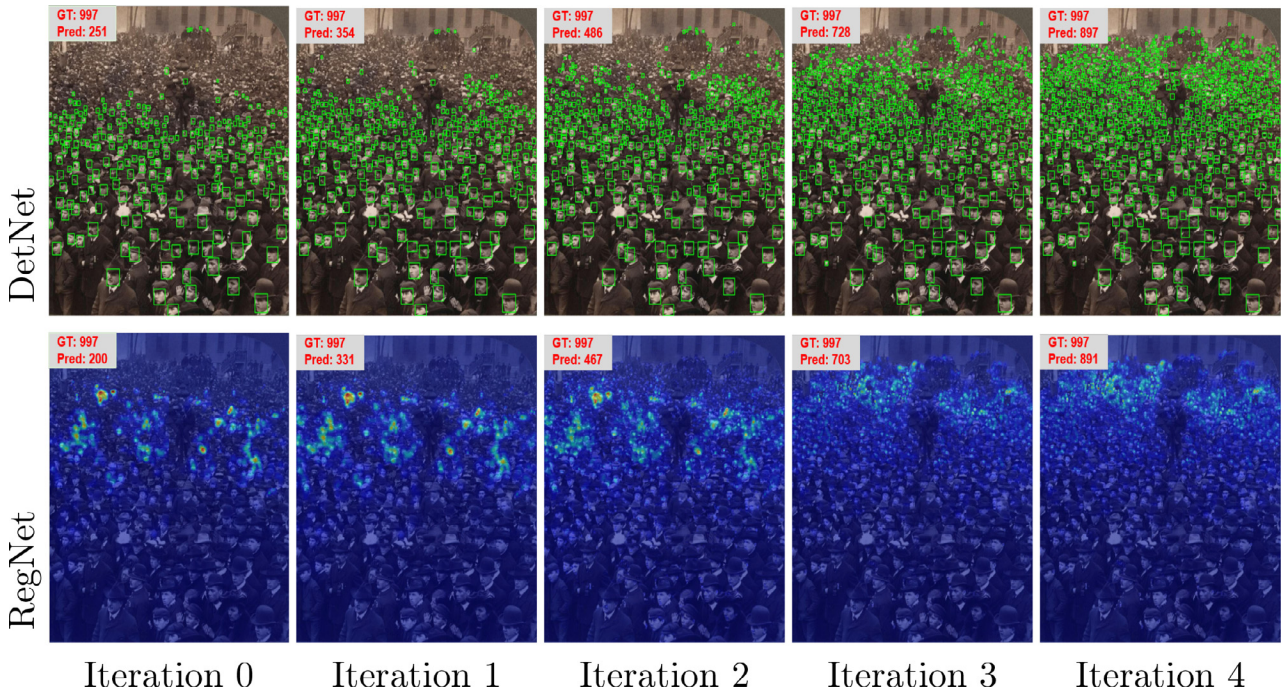


Fig. 4. A visualization example of our method when transferring from SHB to SHA in different iterations.

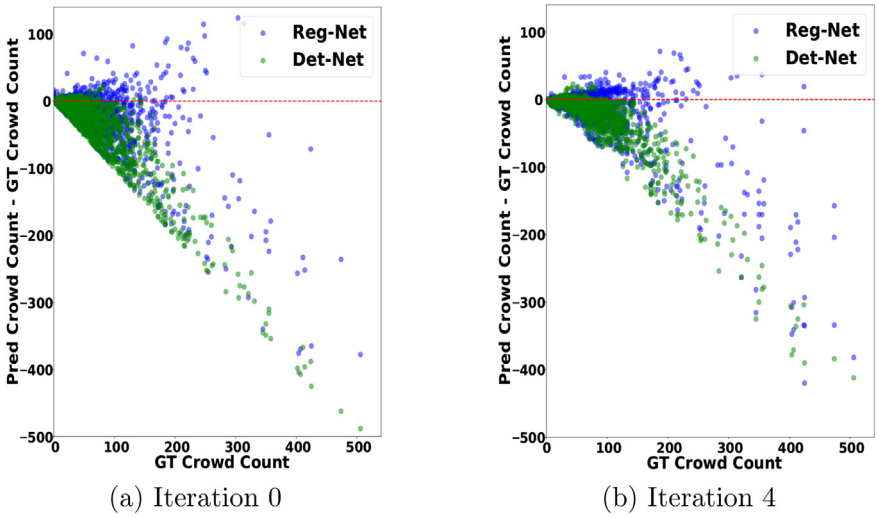


Fig. 5. Comparison of the error crowd count (predicted crowd count - ground truth crowd count) on the ShanghaiTech SHA dataset in Iteration 0 (a) and Iteration 4 (b). The blue and green points show the error distribution by RegNet and DetNet, respectively.

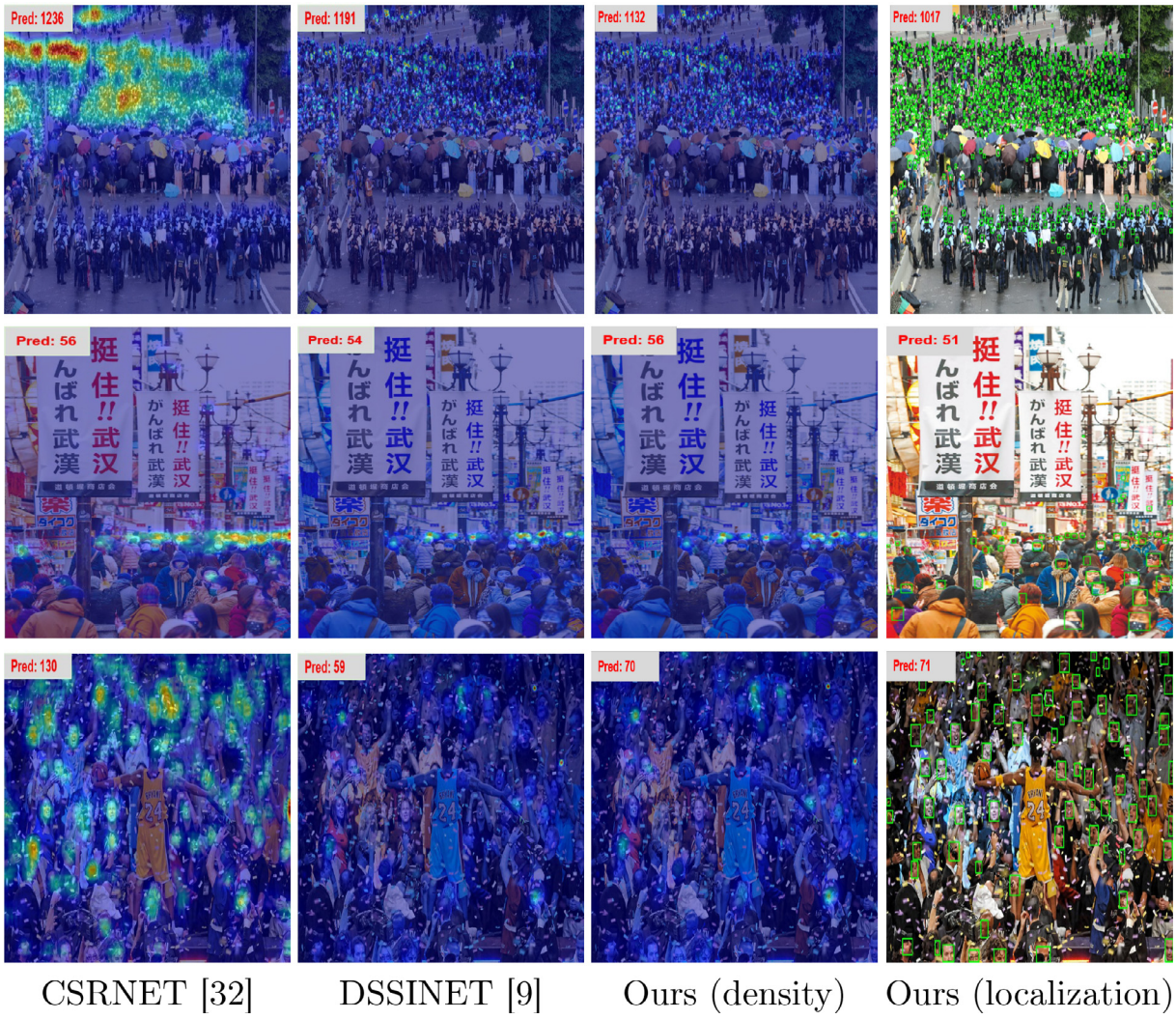


Fig. 6. Visualization results across some unseen scenes. We show the cross-scene results of CSRNET [32], and DSSINET [9] learnt on SHB as well as ours. Our method produces better visual results and more accurate counting results over others.

5. Conclusion

This paper introduced for the first time to discover bi-knowledge transfer between regression and detection models towards unsupervised cross-domain crowd counting. We first model the bi-knowledge transfer between regression and detection models on the source as mutual transformations between the predictions of the two models. Thanks to the modeled scene-agnostic transformers, we let the two models co-teach each other on the target in an iterative self-supervised manner. We further explored a mixup strategy to generate augmented training samples which further enhances the model adaptation. Extensive experiments and analysis clearly demonstrate the effectiveness of our approach. In the future, we will continue to focus on leveraging heterogeneous information from regression- and detection-based models to handle the cross-domain crowd counting issues and dedicate to improving the quality of pseudo labels for unlabelled target data.

CRediT authorship contribution statement

Yuting Liu: Conceptualization, Methodology, Software, Validation, Writing - original draft. **Zheng Wang:** Writing - review & editing. **Miaojing Shi:** Writing - review & editing. **Shin'ichi Satoh:** Supervision. **Qijun Zhao:** Writing - review & editing, Supervision. **Hongyu Yang:** Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The research was partly supported by National Natural Science Foundation of China (No. 62176170, 62066042, 61971005, & 61828602), Sichuan University "Innovation 2035" Pilot Program, JST CREST Grant (JPMJCR1686), and Grant-in-Aid for JSPS Fellows (18F18378).

References

[1] H. Fu, H. Ma, H. Xiao, Crowd counting via head detection and motion flow estimation, in: Proceedings of the ACM International Conference on Multimedia (ACM MM), 2014, pp. 877–880..

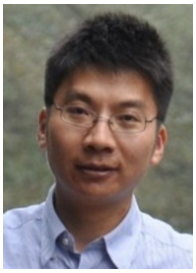
[2] L. Boominathan, S.S. Kruthiventi, R.V. Babu, Crowdnet: A deep convolutional network for dense crowd counting, in: Proceedings of the ACM International Conference on Multimedia (ACM MM), 2016, pp. 640–644..
[3] X. Tan, C. Tao, T. Ren, J. Tang, G. Wu, Crowd counting via multi-layer regression, in: Proceedings of the ACM International Conference on Multimedia (ACM MM), 2019, pp. 1907–1915..
[4] D. Guo, K. Li, Z. Zha, M. Wang, Dadnet: Dilated-attention-deformable convnet for crowd counting, in: Proceedings of the ACM International Conference on Multimedia (ACM MM), 2019, pp. 1823–1832..
[5] Y. Liu, M. Shi, Q. Zhao, X. Wang, Point in, box out: Beyond counting persons in crowds, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 6469–6478..
[6] Z. Cheng, J. Li, Q. Dai, X. Wu, A.G. Hauptmann, Learning spatial awareness to improve crowd counting, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 6152–6161..
[7] A. Zhang, J. Shen, Z. Xiao, F. Zhu, X. Zhen, X. Cao, L. Shao, Relational attention network for crowd counting, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 6788–6797..
[8] Z. Cheng, J. Li, Q. Dai, X. Wu, J. He, A.G. Hauptmann, Improving the learning of multi-column convolutional neural network for crowd counting, in: Proceedings of the ACM International Conference on Multimedia (ACM MM), 2019, pp. 1897–1906..
[9] L. Liu, Z. Qiu, G. Li, S. Liu, W. Ouyang, L. Lin, Crowd counting with deep structured scale integration network, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 1774–1783..
[10] M. Shi, Z. Yang, C. Xu, Q. Chen, Revisiting perspective information for efficient crowd counting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 7279–7288..
[11] Z. Shi, L. Zhang, Y. Liu, X. Cao, Y. Ye, M.-M. Cheng, G. Zheng, Crowd counting with deep negative correlation learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 5382–5390..
[12] C. Xu, K. Qiu, J. Fu, S. Bai, Y. Xu, X. Bai, Learn to scale: Generating multipolar normalized density maps for crowd counting, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 8382–8390..
[13] D. Lian, J. Li, J. Zheng, W. Luo, S. Gao, Density map regression guided detection network for rgb-d crowd counting and localization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 1821–1830..
[14] W. Liu, S. Liao, W. Ren, W. Hu, Y. Yu, High-level semantic feature detection: A new perspective for pedestrian detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5187–5196..
[15] Y. Zhang, D. Zhou, S. Chen, S. Gao, Y. Ma, Single-image crowd counting via multi-column convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 589–597..
[16] J. Liu, C. Gao, D. Meng, A.G. Hauptmann, Decidenet: Counting varying density crowds through attention guided detection and density estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 5197–5206..
[17] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: Proceedings of the International Conference on Learning Representations (ICLR), 2018..
[18] H. Idrees, I. Saleemi, C. Seibert, M. Shah, Multi-source multi-scale counting in extremely dense crowd images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2547–2554..
[19] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, M. Shah, Composition loss for counting, density map estimation and localization in dense crowds, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 532–546..
[20] Y. Liu, Z. Wang, M. Shi, S. Satoh, Q. Zhao, H. Yang, Towards unsupervised crowd counting via regression-detection bi-knowledge transfer, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 129–137..
[21] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision (CVPR), 2017, pp. 2223–2232..
[22] Q. Wang, J. Gao, W. Lin, Y. Yuan, Learning from synthetic data for crowd counting in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 8198–8207..
[23] T. Han, J. Gao, Y. Yuan, Q. Wang, Focus on semantic consistency for cross-domain crowd understanding, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 1848–1852..
[24] L. Wang, Y. Li, X. Xue, Coda: Counting objects via scale-aware adversarial density adaption, in: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), 2019, pp. 193–198..
[25] Q. Wang, T. Han, J. Gao, Y. Yuan, Neuron linear transformation: Modeling the domain shift for crowd counting, IEEE Transactions on Neural Networks and Learning Systems..
[26] D.B. Sam, S. Surya, R.V. Babu, Switching convolutional neural network for crowd counting, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4031–4039..
[27] J. Gao, Q. Wang, Y. Yuan, Scar: Spatial-/channel-wise attention regression networks for crowd counting, Neurocomputing 363 (2019) 1–8.
[28] Z. Zou, Y. Cheng, X. Qu, S. Ji, X. Guo, P. Zhou, Attend to count: Crowd counting with adaptive capacity multi-scale cnns, Neurocomputing 367 (2019) 75–83.
[29] Y. Fang, S. Gao, J. Li, W. Luo, L. He, B. Hu, Multi-level feature fusion based locality-constrained spatial transformer network for video crowd counting, Neurocomputing 392 (2020) 98–107.
[30] X. Wu, Y. Zheng, H. Ye, W. Hu, T. Ma, J. Yang, L. He, Counting crowds with varying densities via adaptive scenario discovery framework, Neurocomputing 397 (2020) 127–138.
[31] V. Lempitsky, A. Zisserman, Learning to count objects in images, in: Advances in Neural Information Processing Systems (NeurIPS), 2010, pp. 1324–1332..
[32] Y. Li, X. Zhang, 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 (CVPR), 2018, pp. 1091–1100..
[33] V.A. Sindagi, V.M. Patel, Multi-level bottom-top and top-bottom feature fusion for crowd counting, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 1002–1012..
[34] J. Wan, A. Chan, Adaptive density map generation for crowd counting, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 1130–1139..
[35] J. Ma, Y. Dai, Y.-P. Tan, Atrous convolutions spatial pyramid network for crowd counting and density estimation, Neurocomputing 350 (2019) 91–101.
[36] L. Yuan, Z. Qiu, L. Liu, H. Wu, T. Chen, P. Chen, L. Lin, Crowd counting via scale-communicative aggregation networks, Neurocomputing 409 (2020) 420–430.
[37] R. Stewart, M. Andriluka, A.Y. Ng, End-to-end people detection in crowded scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2325–2333..
[38] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788..
[39] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: Proceedings of the European Conference on Computer Vision (ECCV), 2016, pp. 21–37..
[40] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (6) (2016) 1137–1149.
[41] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988..
[42] X. Liu, J. van de Weijer, A.D. Bagdanov, Leveraging unlabeled data for crowd counting by learning to rank, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7661–7669..
[43] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556..
[44] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778..
[45] https://ulhpc-tutorials.readthedocs.io/en/latest/cuda/exercises/convolution/ [link]. URL: https://ulhpc-tutorials.readthedocs.io/en/latest/cuda/exercises/convolution/..
[46] Z. Zhou, M.M.R. Siddiquee, N. Tajbakhsh, J. Liang, Unet++: Redesigning skip connections to exploit multiscale features in image segmentation, IEEE Transactions on Medical Imaging 39 (6) (2019) 1856–1867.
[47] A. Neubeck, L. Van Gool, Efficient non-maximum suppression, in: Proceedings of the International Conference on Pattern Recognition (ICPR), Vol. 3, 2006, pp. 850–855..
[48] Z. Zhao, M. Shi, X. Zhao, L. Li, Active crowd counting with limited supervision, in: Proceedings of the European Conference on Computer Vision (ECCV), 2020..
[49] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, Y. Bengio, Manifold mixup: Better representations by interpolating hidden states, in: Proceedings of the International Conference on Machine Learning (ICML), 2019, pp. 6438–6447..
[50] P. Blackburn, J. Bos, Computational semantics, Theoria: An International Journal for Theory, History and Foundations of Science (2003) 27–45.
[51] P. Guo, X. Chen, Image semantic feature analysis, in: Pattern Recognition and Machine Vision, 2010..
[52] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, M. Li, Bag of freebies for training object detection neural networks, arXiv preprint arXiv:1902.04103..
[53] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, Imagenet large scale visual recognition challenge, International Journal of Computer Vision (IJCV) 115 (3) (2015) 211–252.
[54] V.A. Sindagi, V.M. Patel, Generating high-quality crowd density maps using contextual pyramid cnns, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 1861–1870..

Yuting Liu received the B.S. degrees from Nanchang University, China, in 2015. She has been in the Masters and PhD Combined Programs of Sichuan University. Her main research area is computer vision, specifically for challenges in crowd analysis.

Zheng Wang received the B.S. and M.S. degrees from Wuhan University, Wuhan, China, in 2006 and 2008, respectively, and the Ph.D. degree from the National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, Wuhan, China, in 2017. He was a JSPS Fellowship Researcher at Shin'ichi Satoh's Lab, National Institute of Informatics, Japan. He is currently a professor in the School of Computer Science, Wuhan University. His research interests focus on person re-identification and instance search. He received the Best Paper Award at the 15th Pacific-Rim Conference on Multimedia (PCM 2014) and the 2017 ACM Wuhan Doctoral Dissertation Award.

Miaojing Shi received the Ph.D.degree from Peking University in 2015. He was a recognized student with the University of Oxford from 2012 to 2013, and a Visiting Student with INRIA Rennes from 2014 to 2015. He is currently an assistant professor at Department of Informatics, King's College London. His research interests include visual search and computer vision.

Shin'ichi Satoh received the B.E. degree in electronics engineering and the M.E. and Ph.D. degrees in information engineering from the University of Tokyo, Tokyo, Japan, in 1987, 1989, and 1992, respectively. He has been a Full Professor with the National Institute of Informatics, Tokyo, Japan, since 2004. He was a Visiting Scientist with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, from 1995 to 1997. His current research interests include image processing, video content analysis, and multimedia databases.



Qijun Zhao received the BSc and MSc degrees both from Shanghai Jiao Tong University, and the PhD degree from the Hong Kong Polytechnic University. He worked as a post-doc researcher with the Pattern Recognition and Image Processing Lab, Michigan State University from 2010 to 2012. He is currently an professor with the College of Computer Science, Sichuan University. His research interests lie in biometrics, particularly, face perception, fingerprint recognition, and affective computing, with applications to forensics, intelligent video surveillance, mobile security, healthcare, and human–computer interactions. He has published more than 60 papers in academic journals and conferences, and participated in many research projects either as principal investigators or as primary researchers. He is a program committee co-chair of CCBR 2016 and ISBA 2018, and an area co-chair of BTAS 2018. He is a member of the IEEE.

Hongyu Yang is a professor and Ph.D. supervisor at the College of Computer Science, Sichuan University. Her research interests include artificial intelligence, intelligence optimization, air traffic management and image processing.