Variational Attention: Propagating Domain-Specific Knowledge for Multi-Domain Learning in Crowd Counting

Binghui Chen[*], Zhaoyi Yan[*1], Ke Li, Pengyu Li, Biao Wang, Wangmeng Zuo[1†], Lei Zhang[2]
[1] Harbin Institute of Technology, [2] The Hong Kong Polytechnic University
chenbinghui@bupt.cn, yanzhaoyi@outlook.com, like1990@bupt.edu.cn, lipengyu007@gmail.com
wangbiao225@foxmail.com, wmzuo@hit.edu.cn, cslzhang@comp.polyu.edu.hk

Abstract

In crowd counting, due to the problem of laborious labelling, it is perceived intractability of collecting a new large-scale dataset which has plentiful images with large diversity in density, scene, etc. Thus, for learning a general model, training with data from multiple different datasets might be a remedy and be of great value. In this paper, we resort to the multi-domain joint learning and propose a simple but effective Domain-specific Knowledge Propagating Network (DKPNet) for unbiasedly learning the knowledge from multiple diverse data domains at the same time. It is mainly achieved by proposing the novel Variational Attention(VA) technique for explicitly modeling the attention distributions for different domains. And as an extension to VA, Intrinsic Variational Attention(InVA) is proposed to handle the problems of over-lapped domains and sub-domains. Extensive experiments have been conducted to validate the superiority of our DKPNet over several popular datasets, including ShanghaiTech A/B, UCF-QNRF and NWPU.

1. Introduction

Crowd counting is a challenging problem since it suffers from multiple actual issues behind data distributions, such as high variability in scales, density, occlusions, perspective distortions, background scenarios, *etc*. A direct solution to mitigate these issues is to collect a large-scale dataset with abundant data variations like ImageNet[7], so as to encourage the learned model to be more robust and general. However, collecting such a large-scale dataset with rich diversity for crowd-counting training is intractable due to the difficulty in human-labeling. Specifically, in crowd counting, due to the limitation of various conditions, images collected by a research group might only contain certain types of variations and are limited in numbers. For example, as shown in Fig.1, one can observe that there are large variations in data distributions across different datasets. Images in ShanghaiTech A(SHA)[64] tend to show congested crowds, and those in UCF-QNRF(QNRF)[15] are more likely to depict highly-congested crowds and have more background scenarios, and those in NWPU[54] have much more diversities in scales, density, background, *etc*. In contrast, samples within ShanghaiTech B(SHB)[64] just prefer low density crowds and the ordinary street-based scenes. Considering the aforementioned facts, in order to learn a general and robust estimating model for correct density prediction, this paper resorts to *multi-domain learning* which aims to solve the same or similar problem with multiple datasets across different domains[1] simultaneously by utilizing all the data these domains provide. In other words, multi-domain learning gives chances of using relatively abundant data variations coming from different datasets for learning a general and robust density estimating model.

Figure 1: Data distribution comparison between ShanghaiTech [64], UCF-QNRF [15] and NWPU [54]. ShanghaiTech A is mainly composed of congested images, QNRF is of highly-congested samples and have more background scenarios, NWPU covers a much larger variety of data distributions due to density, perspective, background, *etc*, while ShanghaiTech B prefers low density and ordinary street-based scenes.

However, in crowd counting, an interesting phenomenon can be observed when jointly training with multiple different datasets. As shown in Tab.1, if under the supervision of the 3-joint of SHA, SHB and QNRF, the deep model prefers to only improve the performances on SHA and QNRF, while sacrificing that on SHB (5% performance drop). Actually, this kind of phenomenon (i.e. biased/partial learning)

[*]Equal contribution
[†]Corresponding author

[1]Commonly, a *domain* often refers to a data set where samples follow the similar or same underlying data distribution[57].

Table 1: MAE Results of IT/JT in 3-Joint datasets.

Methods	SHA	SHB	QNRF
Individual Training (IT)	60.6	8.8	97.7
Joint Training (JT)	60.2 (↓)	9.3 (↑)	92.8 (↓)

exists widely in computer vision fields. It is because that deep models have the partial/biased learning behavior[2], i.e. *deep models easily learn to focus on surface statistical regularities rather than more general abstract concepts*. In other words, deep models will selectively learn the dominant data knowledge from certain dominant domains[2] while ignoring other potential helpful information from the rest domains. To this end, developing an effective algorithm that can successfully utilize all the knowledge from different datasets remains important.

In this paper, we propose the ***Domain-specific Knowledge Propagating Network*** (DKPNet) for multi-domain joint learning, which intends to refine the propagated knowledge according to the domain-specific distributions and highlight all domains without bias. Specifically, a novel *Variational Attention* (VA) technique is introduced for facilitating the domain-specific attention learning. Based on VA, the output attention distribution can be easily controlled by the latent variable. And we apply the Gaussian Mixture distribution as the prior in the proposed VA for multi-domain learning. Furthermore, as an extension to VA, the *Intrinsic Variational Attention* (InVA) is proposed for handling the potential problems of overlapped-domains and sub-domains. VA and InVA both insist to provide domain-specific guidance for knowledge propagating, but start from coarse and intrinsic perspectives, respectively. In summary, the contributions of this paper are listed as follows:

- DKPNet is proposed to learn a general and robust density estimating model for crowd counting by multi-domain joint learning, which can successfully prevent the model from just learning several dominant domains, and can consistently improve the performances over all the datasets.
- VA/InVA are introduced to provide the domain-specific guidance for refining the propagating knowledge with the help of latent variable. To our knowledge, it is the first work to use variational learning in attention for crowd counting.
- Extensive experiments have been conducted on several popular datasets, including ShanghaiTech A/B[64], UCF-QNRF[15] and NWPU[54], and achieve the state-of-the-art performances on the MAE evaluation.

2. Related Work

Crowd Counting: We review the recent works by the techniques they applied. Such techniques include multi-scale [64, 42, 47, 30, 43], multi-task [63, 14, 27], attention [25, 48, 61, 62, 16, 35], perspective map [44, 59, 60], GNN [31], loss functions [32][52], classification [58] detection [22, 41], NAS [13] and others [47, 46, 24, 38, 26, 51, 55, 49, 45, 28, 1]. However, none of these works pay attention to the multi-domain learning in crowd counting.

Cross-domain Learning in Crowd Counting: Cross-domain learning can be categorized into one/few shot learning [11, 40], domain adaption [20, 55, 8], *etc*. [40] presents a meta-learning inspired approach to solve the few-shot scene adaptive crowd counting problem, and [11] further introduces one-shot scene-specific crowd counting. For domain adaption, CODA [20] performs adversarial training with pyramid patches from both source- and target-domain, so as to tackle different object scales and density distributions. Wang *et al*. [55] release a large synthetic dataset (GCC), and propose SE Cycle GAN to bridge the domain gap between the synthetic and real data. Gao *et al*. [8] propose Multi-level Feature aware Adaptation (MFA) and Structured Density map Alignment (SDA) to extract domain invariant features and produce density maps with a reasonable distribution on the real domain.

Multi-domain Learning: Multi-domain learning aims at improving the performance over multiple domains. And it has been exploited in many fields [36, 57, 39, 9, 29, 56, 19, 5]. Despite of heavy research on multi-domain learning in these fields, there are few corresponding research works in crowd counting. The most relevant work to our method is [34]. [34] presents domain-specific branches embedded behind a fixed ImageNet [7] classification network. And, a domain classifier is proposed to decide which branch to process the input image. As a result, the final performance is not good and limited by the hard assignment of branches. Moreover, the computation cost of this work is in linearly correlation with the number of domains. However, different from this work, DKPNet is light in parameters and computation, and is more flexible and general in both training and testing phases, leading to much better performances.

Variational Learning: VAE[18] has been widely explored and used in generative model families and is good at controlling the output distribution via the latent variable. Based on VAE, conditional-VAE[50] proposes a deep conditional model for structured output prediction using Gaussian latent variables; β-VAE[10] is proposed to use balancing term β to control the capacity and the independence prior; β-TCVAE[6] further extends β-VAE by introducing a total correlation term. All these methods aim at using variational learning for generating visually-good images. However, in this paper, we integrate the variational learning into the attention mechanism, and propose the Variational Attention for learning domain-specific attentions.

[2]Since SHA and QNRF data are much more similar than SHB data, and when combining them three together, SHA and QNRF turn to be the dominant domains.

3. Proposed Method

In this section, we will first give the motivation of our method in Sec.3.1, then introduce the *Variational Attention* (VA) and the *Intrinsic Variational Attention* (InVA) modules in Sec.3.2 and Sec.3.3, *resp*. Finally provide the whole pipeline of ***Domain-specific Knowledge Propagating Network*** (DKPNet) in Sec.3.4.

3.1. Motivation

As experimented in Tab.1, optimizing a deep model by directly employing all the data coming from SHA, SHB and QNRF datasets gives rise to the problem of biased domain learning behavior[2, 4], i.e. the deep model prefers to mainly focusing on the learning of the dominant domains instead of all the domains. This will lead to confusions in model prediction when giving data from non-dominant domains, since these domains are not well learned. Obviously, this phenomenon is unsatisfying and the learned model is not what we want.

Considering the above fact that not all of the useful knowledge from these datasets could be captured, this paper tries to use the *attention mechanism*. ***"Attention" gives chances of capturing the desired information by re-weighting or refining the information/knowledge flow within deep models, so as to enhance the learning ability***. However, the conventional attention modules like SENet[12, 3] are always "*self-attention*" in fact. This will lead to the unconstrained and confused attention distribution outputs for different domains, and when these outputs are used again for re-weighting the original input data, the data distributions will be perturbed, resulting in difficulty of learning especially in multi-domain cases. Therefore, inspired by VAE[18],in order to **control the attention outputs for domains with different distributions, we propose the *Variational Attention* technique**.

3.2. Variational Attention

Without loss of generality, suppose we have multiple datasets $\mathcal{X}^*$ and they have tight and different distributions with each other, each dataset has multiple instances $X_i^* \in \mathcal{X}^*, i \in [1, \ldots, N^*]$, N^* refers to the image number of datasets $\mathcal{X}^*$, and each dataset has been given a coarse label l^* where $l^* \in [0, \ldots, C-1]$ and C is the number of datasets. After feeding each image into the deep model, we can obtain a 3-D tensor x at certain layer for learning attention proposals y.

As aforementioned, in order to control the attention distribution $p_\theta(y)$ where θ is the model parameter, we follow the VAE idea by introducing the latent variable z for controlling the distribution $p_\theta(y)$. Specifically, for modeling different-domain attention distributions, we maximizing the log-likelihood of the conditional probability $(p_\theta(y|x,l))$ as follows:

$$\begin{aligned}\log(p_\theta(y|x,l)) &= \log(\int \frac{p_\theta(y,z|x,l)}{q_\phi(z|x,l)} q_\phi(z|x,l)dz) \\ &\geq \mathbb{E}_{q_\phi(z|x,l)} \log(\frac{p_\theta(y|z,x,l)p_\theta(z|x,l)}{q_\phi(z|x,l)}) \\ &= \mathbb{E}_{q_\phi(z|x,l)} \log(p_\theta(y|z,x,l)) - KL(q_\phi(z|x,l)||p_\theta(z|x,l))\end{aligned} \tag{1}$$

This objective function is the evidence lower bound (ELBO) and includes two terms. The first term tries to maximize the likelihood to improve the confidence of prediction, in other words, it tries to produce good attention proposals so as to benefit the density estimation (in this paper, it corresponds to density estimating loss, which will be described later, $\int_{q_\phi(z|x,l)} \|\hat{Y}(y) - Y\|_2^2 dz$, where Y is the ground-truth of density map and $\hat{Y}(y)$ is estimating result based on the attention output y). The second term refers to the KL divergence between the variational distribution $q_\phi(z|x,l)$ (parameterized by ϕ) and the prior distribution $p_\theta(z|x,l)$, as it is prior distribution we use $p(z|x,l) = p_\theta(z|x,l)$ later. Since the output attention distributions for different domains should be different with each other, we set the prior distribution of the latent variable $z \in \mathbb{R}^d$ to a commonly used Gaussian mixture distribution with C Gaussian components, where C is the number of domains and d is the dimension of z:

$$z \sim \sum_{c=0}^{C-1} \gamma_c \mathcal{N}(u_c, \Sigma_c),\ \forall c, \gamma_c \geq 0, \sum_{c=0}^{C-1} \gamma_c = 1 \tag{2}$$

For each domain, u_c is the corresponding mean vector and Σ_c is the d-dimensional covariance matrix. For convenience, we set $\gamma_c = \frac{1}{C}$ and Σ_c is diagonal matrix throughout this paper. Then, the second term in Eq. 1 can be expressed as:

$$\begin{aligned}KL(q_\phi(z|x,l)||p(z|x,l)) &= \frac{1}{2}[\log(\frac{det(\Sigma_c)}{det(\Sigma_\phi)}) - d \\ &+ tr(\Sigma_c^{-1}\Sigma_\phi) + (u_c - u_\phi)\Sigma_c^{-1}(u_c - u_\phi)^T]\end{aligned} \tag{3}$$

where $l = c$, the parameterized distribution $q_\phi(z|x,l) \sim \mathcal{N}(u_\phi, \Sigma_\phi)$, and u_ϕ, Σ_ϕ are the outputs of model ϕ.

As the $\mathbb{E}_{q_\phi(z|x,l)} log(p_\theta(y|z,x,l))$ is computationally intractable and the total process should be differentiable, we use the reparameterization trick [18] for computation as:

$$\mathbb{E}_{q_\phi(z|x,l)} \log(p_\theta(y|z,x,l)) \simeq \frac{1}{N} \sum_{j=1}^{N} \log(p_\theta(y|z^j,x,l)) \tag{4}$$

where z^j is sampled by $u_\phi + \Sigma_\phi \odot \varepsilon, \varepsilon \sim \mathcal{N}(0, I)$.

As mentioned before, each domain is supposed to have a Gaussian distribution, however, it is hard to factitiously set and fix the prior parameters u_c, Σ_c. We propose to make

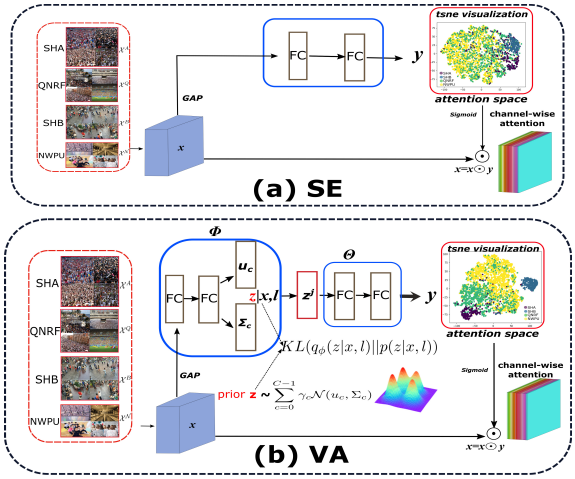

Figure 2: Comparisons between SE and VA. Different datasets are trained jointly. $\odot$ means channel-wise product. One can observe that SE attention outputs are confusing, while our VA can produce more separable attention distributions for different domains by introducing the Gaussian Mixture distributed latent variable z.

them to be the learnable parameters for adaptively adjusting and add a distribution regularizer for getting non-trivial solutions. Considering the semantic relations between domains, we propose to regularize the similarity between the sampled z^j coming from c-th domain and the prior learnable parameters u_c to be the largest among all the similarities between z^j and $u_i, i \in [0, \dots, C-1]$. This can be expressed to force:

$$z^{j^T} u_c \geq \{z^{j^T} u_0, \dots, z^{j^T} u_{C-1}\} \Rightarrow$$
$$\max(z^{j^T} u_0 - z^{j^T} u_c, \dots, z^{j^T} u_{C-1} - z^{j^T} u_c) \leq 0 \quad (5)$$

because the above $\max$ function is piecewise discontinuous function, here, we minimize its upper-bound function log-sum-exp instead for optimization:

$$L_{reg} = \log(\sum_{i=0}^{C-1} e^{z^{j^T} u_i - z^{j^T} u_c}) \geq \max(z^{j^T} u_0 - z^{j^T} u_c, \dots, z^{j^T} u_{C-1} - z^{j^T} u_c) \quad (6)$$

Minimizing Eq.6 can help regularizing different domains to have different distributions. And for Σ_c, we simply regularize it by minimizing $\log^2(det(\Sigma_c))$, and add it to L_{reg}. Finally, VA can be optimized by minimizing the following loss:

$$L_{VA} = \frac{-1}{N} \sum_{j=1}^{N} \log(p_\theta(y|z^j, x, l)) + KL(q_\phi(z|x,l)||p(z|x,l)) + \log(\sum_{i=0}^{C-1} e^{z^{j^T} u_i - z^{j^T} u_c}) + \log^2(det(\Sigma_c)) \quad (7)$$

and the attention output y will be further used to re-weight the input tensor x as $x = x \odot y$.

Remark: By introducing and modeling the latent variable, the output attention will be domain-related, such that the domain-specific knowledge can be well captured and learned in multi-domain cases. As shown in Fig. 2, different from SE attention, our VA can produce more separable attention distributions for different domains with the help of latent variable modeling. And the latent variable is assumed to be Gaussian Mixture distribution, such that for each domain, an independent Gaussian distribution can be applied and used to control the attention outputs.

3.3. Intrinsic Variational Attention

Actually, the above VA simply assumes that each dataset belongs to a single domain. However, it does not hold on in many cases and there might be two frequent problems: (1) *domain-overlaps* across different datasets and (2) *sub-domains* within the same dataset. Therefore, labels l will be too coarse to provide the fine-grained and exact guidance for domain-specific attention learning, still leaving behind some confusions in attention learning. To this end, in order to capture the intrinsic domain labels for accurate domain-specific attention learning, we extend our VA into ***Intrinsic Variational Attention*** (InVA). It is mainly achieved by using *Clustering*(CL) labels and *Sub-Gaussian Components*(SGC) which intend to mitigate the problems of domain-overlaps and sub-domains, respectively.

Specifically, in order to tackle the domain-overlaps, it is required to reassign the more correct domain-labels to the training data instead of using the original dataset-label l. Thus, we first train a VA and then perform the Gaussian-Mixture clustering[3] over the attention proposals outputed by the VA module. The number of clusters is set to $\bar{C}$. After clustering, the new domain labels $\bar{l} \in [0, \dots, \bar{C}-1]$, i.e. CL labels, are reassigned to the original training data.

Moreover, there also might be sub-domains within each clustered domain since the clustering is unsupervised and not capable of finding each potential sub-domains. In order to cope with the potential sub-domains, we propose to use SGC for the latent variable modeling. Specifically, we assume there are at most k sub-domains in each clustered domain. Thus, the latent variable $z \in \mathbb{R}^d$ will turn to be a Sub-Gaussian Mixture distribution as follows:

$$z \sim \sum_{c=0}^{\bar{C}-1} \frac{1}{\bar{C}} \mathcal{N}(u_c, \Sigma_c)$$
$$where\ u_c = \arg\max_{u_{c,i}} \{\sigma(u_{c,1}^T u_\phi), \dots, \sigma(u_{c,k}^T u_\phi)\} \quad (8)$$

σ means Dropout with drop-rate 0.2, $u_{c,k}$ is center vector of the k-th sub-component in the c-th Gaussian. For simplicity, we use the same Σ_c for the sub-gaussian components. Finally, using the CL labels $\bar{l}$ and substituting the

[3] We also tested other clustering methods, e.g. Kmeans, DBSCAN, *etc*, the performances are similar.

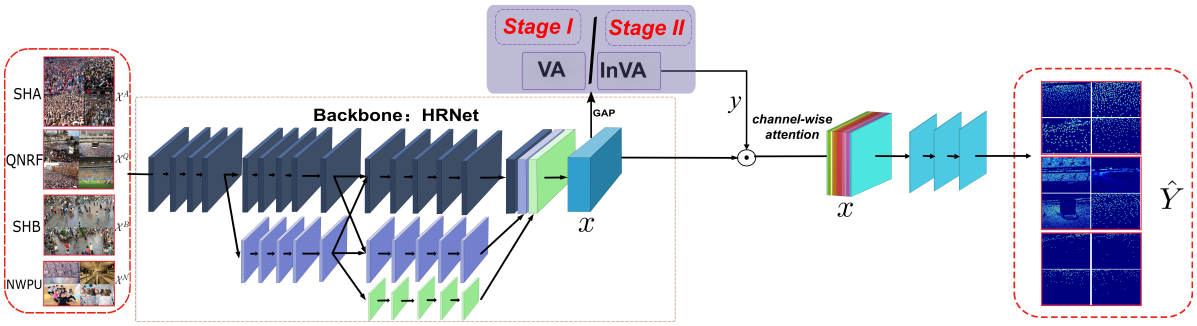

Figure 3: The pipeline of the proposed DKPNet. It contains two-stage training: VA and InVA are used in Stage-I and Stage-II, respectively. Before Stage-II training, we will first get the CL labels from the attention outputs of VA by clustering, and then reassign the training images with these CL labels.

latent variable z in Eq. 8 into Eq. 7 can obtain the loss function L_{InVA} for training InVA.

Remark: CL labels focus on handling the overlapped-domains across different datasets. SGC allows the existence of sub-domains in each clustered domain and is capable of adaptively optimizing these sub-domains. Thus, equipping with both CL labels and SGC, InVA module can provide more accurate guidance for knowledge refining by attention according to the "*intrinsic*" domains of data. As the basic module structure is similar with VA except for the CL labels and prior distribution of z, here for the limitation of paper length, we omit the figure for showing InVA.

3.4. DKPNet

Now, we will introduce the whole pipeline of our DKPNet as in Fig.3. Here, we take the truncated HRNet[53](for parameter lightness, we only use parameters from *stage1* to *stage3*) as our backbone. During training, the mini-batched images are randomly sampled from all datasets and then are fed into the backbone together. A 1×1 convolution with 512 channels is applied after, producing a 4-D tensor x. Then tensor x will be passed to VA/InVA for producing the domain-specific attention weights y, which will be applied on x by channel-wise attention, resulting in a new tensor $x = x \odot y$. Notably, VA/InVA are used in stage-I and stage-II, respectively. And for each stage training, the backbone is initialized by the ImageNet-pretrained model. Finally, for predicting the density maps $\hat{Y}(y)$, three 1×1 convolution layers (with channels $64, 32, 1$, respectively) are adopted. After replacing $-\log(p_\theta(y|z^j, x, l))$ by the density estimating loss $\|\hat{Y}(y)_i - Y_i\|_2^2$, the whole DKPNet can be optimized by the following objective function:

$$L = \frac{1}{2B}\sum_{i=1}^{B}\|\hat{Y}(y)_i - Y_i\|_2^2 + KL(q_\phi(z|x,l)||p(z|x,l)) + \log(\sum_{i=0}^{C-1} e^{z^{j\,T}u_i - z^{j\,T}u_c}) + \log^2(det(\Sigma_c)) \quad (9)$$

B is the batch size and for VA/InVA, the latent variable z are modeled by Eq. 2 and Eq. 8, *resp*.

Remark: VA/InVA are performed in order, and progressively aim at refining the propagating information flows by attention, such that the data from different distributions can be unbiasedly treated and learned, without inducing confusions in predictions.

4. Experiments

Datasets: We conduct experiments on ShanghaiTech A/B[64], UCF-QNRF[15] and NWPU[54]. SHA contains 482 crowd images with crowd numbers varying from 33 to 3139, where 300 images are used for training and the rest 182 images are used for testing. SHB contains 716 images with crowd numbers varying from 9 to 578, where 400 images are employed for training and the rest 316 images are for testing. QNRF [15] contains $1,535$ images. These images are split into the training set with $1,201$ images and the testing set with 334 images, respectively. This dataset has much more annotated heads and prefers highly-congested density. NWPU dataset[54] is a new public dataset which consists of $5,109$ images, including $3,109$ training images, 500 val images and $1,500$ test images, where the test images can only be evaluated on the official website.

Mention & Notations: As in *multi-domain* joint learning, these datasets are trained simultaneously by a single model, and tested individually, i.e. reporting the results on each datasets separately. "DKPNet(c, k)" denotes that we use c clustered domains and at most k sub-domains in the proposed DKPNet.

Implementation details: The proposed DKPNet is applied on the truncated HRNet-W40[53] architecture which is pretrained on ImageNet[7]. For model training, we adopt the Adam[17] optimizer with default betas$= (0.9, 0.999)$, set the start learning rate to be 0.00005 for both the pretrained backbone and the new added layers, and use a total of 450 epochs with the learning rate decreased by a factor of 2.5 per 250 epochs. For data preprocessing, we adopt fixed Gaussian kernels with size 15 to generate the ground-truth density maps. For images with the shortest side smaller than 416, we will resize the shortest side to 416 by keeping the aspect ratio. Then during training, random-cropping with

Table 2: Results on SHA, SHB, QNRF and NWPU. "Individual" means the models are trained by only one individual dataset. "3-Joint" and "4-Joint" refer to using the joint dataset of (SHA,SHB,QNRF) and (SHA,SHB,QNRF,NWPU), respectively. For each joint dataset, only a single model is trained. Moreover, "IT" refers to training models just with the individual datasets. "JT" means merging all the datasets for training. For fair comparison, IT and JT are performed under the same training settings as our DKPNet. The best results are in red color. NWPU(V) and NWPU(T) indicate Val and Test sets on NWPU, *resp*.

Methods	SHA[64]		SHB[64]		QNRF[15]		NWPU[54] (V)		NWPU[54] (T)	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Training Dataset: Individual										
CSRNet [21]	68.2	115.0	10.6	16.0	-	-	104.8	433.4	121.3	387.8
CANet [26]	62.3	100.0	7.8	12.2	107.0	183.0	93.5	489.9	106.3	386.5
SFCN [55]	64.8	107.5	7.6	13.0	102.0	171.0	95.4	608.3	105.4	424.1
DSSINet [23]	60.6	96.0	6.9	10.3	99.1	159.2	-	-	-	-
Bayes [32]	62.8	101.8	7.7	12.7	88.7	154.8	93.6	470.3	105.4	454.2
DM-Count [52]	59.7	95.7	7.4	11.8	85.6	148.3	70.5	357.6	88.4	388.6
IT(baseline1)	60.6	99.2	8.8	12.6	97.7	155.7	81.7	516.0	94.0	371.9
Training Dataset: 3-Joint										
JT(baseline2)	60.2	99.6	9.3	13.7	92.8	159.7	-	-	-	-
MB [34]	59.4	101.2	8.3	13.2	91.9	159.6	-	-	-	-
DKPNet (c=3,k=3)	***56.7***	***97.1***	***6.9***	***12.0***	***85.2***	***151.4***	-	-	-	-
Training Dataset: 4-Joint										
JT(baseline2)	59.9	96.7	9.7	15.2	91.1	160.4	73.2	509.5	81.9	351.5
MB [34]	59.2	97.7	8.9	13.4	90.6	157.1	72.7	504.0	80.5	377.8
DKPNet (c=5,k=2)	***55.6***	***91.0***	***6.6***	***10.9***	***81.4***	***147.2***	***61.8***	***438.7***	***74.5***	***327.4***

size 400×400, random horizontal flipping and color jittering are adopted. We set the batch size to 32 in all experiments and use two NVIDIA-V100 GPUs. DKPNet is implemented by Pytorch[37] framework.

Evaluation Metrics: We adopt MAE and MSE metrics for evaluations on crowd counting datasets, which is consistent with previous work[21].

4.1. Comparison with State-of-the-Arts

In order to highlight the significance of the proposed DKPNet, we compare it with some recent remarkable works over the popular challenging benchmarks, inlcuding ShanghaiTech A/B[64], UCF-QNRF[15] and NWPU[54]. IT and JT are conducted with the same training configurations as our DKPNet, and both of them are set as our baselines.

More Data & More Better ? To answer this question, we provide some results as in Tab.2. From this table, one can observe that merging[4] more datasets for training a robust estimating model is infeasible. For example (1) in 3-joint case, comparing JT with IT, *JT will biasedly sacrifice the performances on SHB*, i.e. raising the MAE from 8.8 to 9.3 (the same phenomenon can also be observed in the 4-joint datasets training case, i.e. JT raises the MAE on SHB from 8.8 to 9.7). (2) comparing JT results in 3-Joint case with those in 4-Joint case, one can observe that the performances are similar in both 3-Joint and 4-Joint cases, and even the performance on SHB in 4-Joint case is worse than that in 3-Joint case. From the above observations, we can get conclusion that ***using more datasets cannot easily result in a much better model***.

Effectiveness of DKPNet: To this end, DKPNet is proposed and applied in both 3-joint and 4-joint cases. As in Tab.2, one can observe that DKPNet can successfully use more data for learning a better model, i.e. it can consistently improve the performances over both IT and JT baselines. For example, (1) when using 3-Joint(or 4-Joint) datasets DKPNet can surpass the baseline IT by a large margin(e.g. obtaining 61.8 MAE on NWPU(V) with 24% gains); (2) when using 4-Joint datasets, DKPNet can further improve the performances over DKPNet in 3-Joint case. These results verify the importance of our DKPNet for propagating domain-specific knowledge by using the latent-variable-constrained attention. Moreover, DKPNet(c=5,k=2) outperforms all the listed methods in MAE evaluation by a large margin, demonstrating the effectiveness of our DKPNet for multi-domain learning in crowd counting. Notably, DKPNet requires only **O**ne model for all the evaluations, while other methods have to require the corresponding trained model for each dataset. And the visualizations of density maps of JT and DKPNet are in Fig.4.

Comparison with Multi-Branch Learning: Moreover, comparing with the most related work MB[34] which uses a shared backbone followed by multiple branches for processing the different datasets(in which the parameter number and computational cost will increase with the number of datasets in a linear manner), DKPNet can surpass it by a large margin, as shown in Tab. 2, with negligible increases of parameters and computational costs. And MB uses

[4]In JT, we have used the balanced-sampling strategy for different datasets.

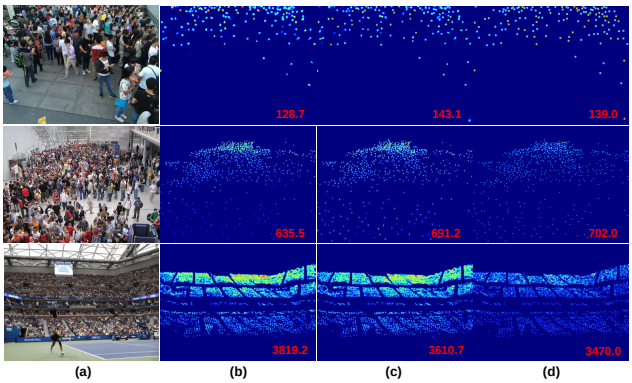

Figure 4: Visualizations of test samples in the 4-Joint case. (a), (b), (c) and (d) are the input, density maps of JT, DKPNet(c=5,k=2) and Ground-Truth, *resp*.

the hard assignment of branches which is too "arbitrary" and cannot handle the problems of overlapped-domains and sub-domains, leading to a few performance improvements over the baseline JT. In contrast, DKPNet is much softer for handling the different data domains by using latent variable constrained attention, and is capable of handling these above problems.

4.2. Component Analysis

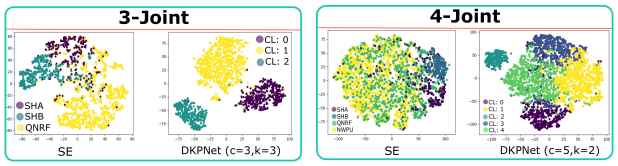

Figure 5: Tsne[33] visualizations of the attention outputs of SE[12] and our DKPNet. Different colors refer to different domains based on the GT labels and CL labels.

Table 3: MAE comparisons between SE[12] and DKPNet. For 3-Joint and 4-Joint cases, we use DKPNet(c=3,k=3) and DKPNet(c=5,k=2), respectively.

Methods	SHA	SHB	QNRF	SHA	SHB	QNRF	NWPU(V)
IT	60.6	8.8	97.7	60.6	8.8	97.7	81.7
	3-Joint			**4-Joint**			
SE[12]	58.1	9.4	88.1	58.0	9.6	97.8	66.4
DKPNet	**56.7**	**6.9**	**85.2**	**55.6**	**6.6**	**81.4**	**61.8**

Why the domain-specific attention works? In order to answer this question and to demonstrate the effectiveness of our enhanced attention method in DKPNet, we take the SE[12] attention for comparison. For fair comparison, we train the SE-based model via adopting the same training configurations and backbone as our DKPNet, only replacing the VA/InVA modules with the SE attention module. We first provide the comparisons of attention spaces via tsne visualization as in Fig.5. Specifically, as in the 3-Joint case, when training with the SE module, the attention distributions for different datasets are very close to each other and even confuse with each other, e.g. the attention weights for SHB are very close to those for SHA and QNRF, even overlaps with them. However, this confusing attention output does not bring the performance improvements on the SHB dataset (i.e. as shown in Tab.3, MAE result on SHB will be weakened from 8.8 to 9.4). It is because that the deep model will produce the accurate predictions if and only if it can exactly capture the true data distributions and then specifically learn knowledge from them. Therefore, when employing the SE attention which has no explicit domain guidance, the images in SHB (which have clear distribution differences with images in (SHA, QNRF)) will be wrongly assigned with the (SHA, QNRF)-like attention weights, resulting in further confusion in model prediction. Moreover, the same phenomenon can also be observed in the 4-Joint case, i.e. SE produces much more confused attention distributions, meanwhile it results in the performance drops on both SHB and QNRF, i.e. drop from 8.8 to 9.6 and from 97.7 to 97.8, *resp*.

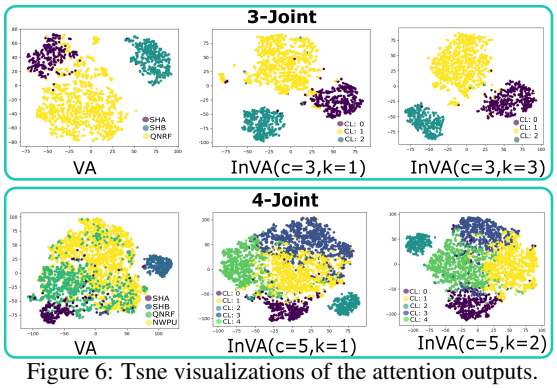

Figure 6: Tsne visualizations of the attention outputs.

Table 4: The MAE results comparisons of VA/InVA.

3-Joint				4-Joint				
Methods	SHA	SHB	QNRF	Methods	SHA	SHB	QNRF	NWPU(V)
IT	60.6	8.8	97.7	IT	60.6	8.8	97.7	81.7
JT	60.2	9.3	92.8	JT	59.9	9.7	91.1	73.2
VA	57.5	7.6	87.9	VA	57.6	7.2	87.6	66.2
InVA(c=3)	57.3	7.5	86.7	InVA(c=5)	56.5	7	84	63.9
InVA(c=3,k=3)	**56.7**	**6.9**	**85.2**	InVA(c=5,k=2)	**55.6**	**6.6**	**81.4**	**61.8**

However, in contrast, as shown in Fig.5 and Tab.3, when training with our domain-specific attention modules, the output attention spaces are much more separable than those outputted by SE. As a result, the propagating information can be handled specifically and exactly for producing correct predictions. Finally, the performances on all datasets can be consistently improved without biases, outperforming SE by a large margin. These phenomenons demonstrate the necessity and importance of explicitly learning the domain-specific attention for multi-domain joint training.

Effect of the VA and InVA: As in Tab.4 and Fig.6, we conduct the quantitative and qualitative comparisons on DKPNet. For convenience, we will take the 3-Joint case for major description. Specifically, the VA tries to learn a coarsely domain-specific attention output via the latent variable z. This leads to the consistent performance improvements over both the baselines IT and JT (e.g. 57.5 vs. 60.2/60.6, 7.6 vs. 9.3/8.8 and 87.9 vs. 92.8/97.7 on SHA, SHB and QNRF respectively.). The learned attention distributions for different datasets are relatively separable ex-

Table 5: MAE results on the value of (c, k) in the 3-Joint case. For convenience, we will omit writing k when $k = 1$.

Methods	SHA	SHB	QNRF	Methods	SHA	SHB	QNRF
InVA(c=2)	**57.1**	7.9	87.5	InVA(c=3,k=2)	57.1	7.3	85.9
InVA(c=3)	57.3	**7.5**	**86.7**	InVA(c=3,k=3)	**56.7**	**6.9**	**85.2**
InVA(c=4)	59.6	7.7	88.2	InVA(c=3,k=4)	57.7	7.4	86.2

cept for some special cases (some overlapped distributions). Moreover, considering that the dataset-labels are not the exact definitions of the intrinsic data domains, we propose the InVA module to further explore the intrinsic domains by data clustering and fine-grained Gaussian Mixture distribution modeling. For example, when using InVA(c=3), one can observe that the aforementioned overlapped distributions can be properly handled to some extent, obtaining further consistent performance improvements over VA (e.g. improving from 57.5 to 57.3, from 7.6 to 7.5 and from 87.9 to 86.7 on SHA, SHB and QNRF, *resp*.). Furthermore, InVA(c=3,k=3) is proposed to cope with the potential sub-domains within each clustered domain. One can observe that after giving chances of learning the potential sub-domains, the attention distributions of each clustered domains are more compact than before and the quantitative performances are further improved(e.g. improving the MAE from 57.3 to 56.7, 7.5 to 6.9 and 86.7 to 85.2 on SHA, SHB and QNRF, *resp*). And the similar performance improvements can also be observed in the 4-Joint case.

In summary, DKPNet concentrates on progressively learning the domain-specific guided information flow by the two-staged training framework, where VA and InVA are performed in order.

Ablation Study on the value of (c, k): As mentioned before, in InVA, we will first obatin the clustered domains that are separable to some extent in the macro-perspective, and then handle the potential sub-domains. The experimental results are shown in Tab.5, one can observe that, for the 3-Joint case, $c = 3$ works the best. It is reasonable since from a global view of the attention space output by VA (see Fig.6), $c = 3$ can well separate the different distributions without much confusion. And for each clustered domain, $k = 2, 3, 4$ works better than $k = 1$ since the potential sub-domains can be explicitly learned, and we experimentally find $k = 3$ is the best. Moreover, for the 4-Joint case, we experimentally find $(c = 5, k = 2)$ works the best.

Sub-domain analysis: In order to explicitly show the learning results of the sub-domains, we compute the cosine similarities between the center vectors of sub-domains (parameterized by $u_{c,k}$) and also calculate out the number of images in each sub-domains as shown from Tab.6-Tab.7. For example, from Tab.6, one can observe that the sub-domains in each clustered domain are specifically learned and different with each other since $\text{Sim}(q, p)$ shows there exists angle between sub-domain centers. Moreover, from Tab.7, it can be observed that all the sub-domains have its corresponding images, implying that the sub-domains indeed exist and are successfully learned.

Table 6: Cosine similarities between sub-domain centers for the 3-Joint case. $\text{Sim}(q, p)$ means the cosine similarity between sub-centers $u_{c,q}$ and $u_{c,p}$.

	CL-0	CL-1	CL-2
$\text{Sim}(0, 1)$	0.61	0.42	0.90
$\text{Sim}(0, 2)$	0.75	0.72	0.78
$\text{Sim}(1, 2)$	0.67	0.59	0.80
avg-sim	0.68	0.58	0.83

Table 7: The number of images in different sub-domains for the 3-Joint case. "Sub-p" is the p-th sub-domain.

	CL-0	CL-1	CL-2
Sub-0	180	423	56
Sub-1	218	86	257
Sub-2	92	510	79

Ablation Study on regularization term: As shown in Tab. 8, comparing with DKPNet, when training DKPNet without regularizing the learning of the prior distribution of z, the performances will be weakened a little. This shows the importance of term L_{reg} for improving the learning of domain-specific attentions by regularizing the differences across domain distributions.

Table 8: Ablation study on regularization term.

Methods	SHA	SHB	QNRF	NWPU(V)
JT(baseline)	59.9	9.7	91.1	73.2
DKPNet w/o L_{reg}	57.3	7.5	86.9	64.2
DKPNet	**55.6**	**6.6**	**81.4**	**61.8**

Model size: From Tab.9, one can observe that the total parameter number of DKPNet is fewer than [32, 52], but DKPNet can surpass them by a large margin, showing that DKPNet is indeed "light and sweet". Moreover, DKPNet only needs one single model for all data evaluations, while [32, 52] have to train many corresponding models for evaluations which are heavy and complicated.

Table 9: Model size comparisons. PN means Parameter Number(million). DKPNet is trained by the 4-Joint datasets.

Methods	PN	SHA	SHB	QNRF	NWPU(V)	NWPU(T)
Bayes[32]	20	62.8	7.7	88.7	93.6	105.4
DM-count[52]	20	59.7	7.4	85.6	70.5	88.4
DKPNet	**14**	**55.6**	**6.6**	**81.4**	**61.8**	**74.5**

5. Conclusion

In this paper, we propose DKPNet for learning the robust and general density estimating model for crowd counting by multi-domain joint learning. Specifically, DKPNet is a two-stage training framework, where the VA module is used in Stage-I for coarsely guiding the domain-specific attention learning, and the InVA module is employed in Stage-II for exploring the intrinsic domains by handling both the problems of overlapped-domains and sub-domains, so as to provide more accurate guidance for domain-specific attention learning. Finally, extensive experiments have been conducted on four popular benchmarks to validate the necessity and effectiveness of our method.

Acknowledgments: We hereby give specifical thanks to Alibaba Group for their contribution to this paper

References

[1] Shuai Bai, Zhiqun He, Yu Qiao, Hanzhe Hu, Wei Wu, and Junjie Yan. Adaptive dilated network with self-correction supervision for counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[2] Binghui Chen and Weihong Deng. Energy confused adversarial metric learning for zero-shot image retrieval and clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8134–8141, 2019.

[3] Binghui Chen and Weihong Deng. Hybrid-attention based decoupled metric learning for zero-shot image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] Binghui Chen, Weihong Deng, and Jiani Hu. Mixed high-order attention network for person re-identification. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[5] Binghui Chen, Weihong Deng, and Haifeng Shen. Virtual class enhanced discriminative embedding learning. In *Advances in Neural Information Processing Systems*, pages 1946–1956, 2018.

[6] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[8] Junyu Gao, Qi Wang, and Yuan Yuan. Feature-aware adaptation and structured density alignment for crowd counting in video surveillance. *arXiv preprint arXiv:1912.03672*, 2019.

[9] Hengkai Guo, Tang Tang, Guozhong Luo, Riwei Chen, Yongchen Lu, and Linfu Wen. Multi-domain pose network for multi-person pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision*, pages 0–0, 2018.

[10] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[11] Mohammad Asiful Hossain, Mahesh Kumar, Mehrdad Hosseinzadeh, Omit Chanda, and Yang Wang. One-shot scene-specific crowd counting. In *BMVC*, page 217, 2019.

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[13] Yutao Hu, Xiaolong Jiang, Xuhui Liu, Baochang Zhang, Jungong Han, Xianbin Cao, and David Doermann. Nas-count: Counting-by-density with neural architecture search. *arXiv preprint arXiv:2003.00217*, 2020.

[14] S. Huang, X. Li, Z. Zhang, F. Wu, S. Gao, R. Ji, and J. Han. Body structure aware deep crowd counting. *IEEE Transactions on Image Processing*, 27(3):1049–1059, March 2018.

[15] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European Conference on Computer Vision*, pages 532–546, 2018.

[16] Xiaoheng Jiang, Li Zhang, Mingliang Xu, Tianzhu Zhang, Pei Lv, Bing Zhou, Xin Yang, and Yanwei Pang. Attention scaling for crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4706–4715, 2020.

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

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] Wenyu Li, Tianchu Guo, Pengyu Li, Binghui Chen, Biao Wang, Wangmeng Zuo, and Lei Zhang. Virface: Enhancing face recognition via unlabeled shallow data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14729–14738, June 2021.

[20] Wang Li, Li Yongbo, and Xue Xiangyang. Coda: Counting objects via scale-aware adversarial density adaption. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 193–198. IEEE, 2019.

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

[22] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander 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*, pages 5197–5206, 2018.

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

[24] Lingbo Liu, Hongjun Wang, Guanbin Li, Wanli Ouyang, and Liang Lin. Crowd counting using deep recurrent spatial-aware network. *arXiv preprint arXiv:1807.00601*, 2018.

[25] Ning Liu, Yongchao Long, Changqing Zou, Qun Niu, Li Pan, and Hefeng Wu. Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3225–3234, 2019.

[26] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5099–5108, 2019.

[27] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7661–7669, 2018.

[28] Yuting Liu, Miaojing Shi, Qijun Zhao, and Xiaofang Wang. Point in, box out: Beyond counting persons in crowds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6469–6478. IEEE, 2019.

[29] Yajing Liu, Xinmei Tian, Ya Li, Zhiwei Xiong, and Feng Wu. Compact feature learning for multi-domain image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7193–7201, 2019.

[30] Miaojing Shi* Lu Zhang* and Qiaobo Chen. Crowd counting via scale-adaptive convolutional neural network. 2018.

[31] Ao Luo, Fan Yang, Xin Li, Dong Nie, Zhicheng Jiao, Shangchen Zhou, and Hong Cheng. Hybrid graph neural networks for crowd counting. *arXiv preprint arXiv:2002.00092*, 2020.

[32] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[33] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[34] Mark Marsden, Kevin McGuinness, Suzanne Little, Ciara E Keogh, and Noel E O'Connor. 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*, pages 8070–8079, 2018.

[35] Yunqi Miao, Zijia Lin, Guiguang Ding, and Jungong Han. Shallow feature based dense attention network for crowd counting. In *AAAI*, pages 11765–11772, 2020.

[36] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[37] Pytorch. https://pytorch.org/.

[38] Viresh Ranjan, Hieu Le, and Minh Hoai. Iterative crowd counting. In *Proceedings of the European Conference on Computer Vision*, pages 270–285, 2018.

[39] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516, 2017.

[40] Mahesh Kumar Krishna Reddy, Mohammad Hossain, Mrigank Rochan, and Yang Wang. Few-shot scene adaptive crowd counting using meta-learning. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2814–2823, 2020.

[41] Deepak Babu Sam, Skand Vishwanath Peri, Mukuntha Narayanan Sundararaman, Amogh Kamath, and Venkatesh Babu Radhakrishnan. Locate, size and count: Accurately resolving people in dense crowds via detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[42] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. Switching convolutional neural network for crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 6, 2017.

[43] Zan Shen, Yi Xu, Bingbing Ni, Minsi Wang, Jianguo Hu, and Xiaokang Yang. Crowd counting via adversarial cross-scale consistency pursuit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5245–5254, 2018.

[44] Miaojing Shi, Zhaohui Yang, Chao Xu, and Qijun Chen. Revisiting perspective information for efficient crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7279–7288, 2019.

[45] Zenglin Shi, Pascal Mettes, and Cees GM Snoek. Counting with focus for free. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4200–4209, 2019.

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

[47] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1879–1888. IEEE, 2017.

[48] Vishwanath A Sindagi and Vishal M Patel. Ha-ccn: Hierarchical attention-based crowd counting network. *IEEE Transactions on Image Processing*, 29:323–335, 2019.

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

[50] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491, 2015.

[51] Jia Wan and Antoni Chan. Adaptive density map generation for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1130–1139, 2019.

[52] Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai. Distribution matching for crowd counting. *arXiv preprint arXiv:2009.13077*, 2020.

[53] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[54] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting. *arXiv preprint arXiv:2001.03360*, 2020.

[55] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8198–8207, 2019.

[56] Jin Xiao, Shuhang Gu, and Lei Zhang. Multi-domain learning for accurate and few-shot color constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2020.

[57] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1249–1258, 2016.

[58] Haipeng Xiong, Hao Lu, Chengxin Liu, Liu Liang, Zhiguo Cao, and Chunhua Shen. From open set to closed set: Counting objects by spatial divide-and-conquer. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[59] Zhaoyi Yan, Yuchen Yuan, Wangmeng Zuo, Xiao Tan, Yezhen Wang, Shilei Wen, and Errui Ding. Perspective-guided convolution networks for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 952–961, 2019.

[60] Yifan Yang, Guorong Li, Zhe Wu, Li Su, Qingming Huang, and Nicu Sebe. Reverse perspective network for perspective-aware object counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4374–4383, 2020.

[61] Anran Zhang, Jiayi Shen, Zehao Xiao, Fan Zhu, Xiantong Zhen, Xianbin Cao, and Ling Shao. Relational attention network for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6788–6797, 2019.

[62] Anran Zhang, Lei Yue, Jiayi Shen, Fan Zhu, Xiantong Zhen, Xianbin Cao, and Ling Shao. Attentional neural fields for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5714–5723, 2019.

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

[64] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 589–597, 2016.