Crowd Counting With Partial Annotations in an Image

Yanyu Xu*[1], Ziming Zhong*[2], Dongze Lian[2], Jing Li[2], Zhengxin Li[2], Xinxing Xu[1]
Shenghua Gao† [2,3,4]

[1]IHPC, A*STAR, Singapore. {xu_yanyu,xuxinx}@ihpc.a-star.edu.sg
[2]ShanghaiTech University, China. {zhongzm,liandz,lijing1,lizhx, gaoshh}@shanghaitech.edu.cn
[3]Shanghai Engineering Research Center of Intelligent Vision and Imaging, China.
[4]Shanghai Engineering Research Center of Energy Efficient and Custom AI IC, China.

Abstract

To fully leverage the data captured from different scenes with different view angles while reducing the annotation cost, this paper studies a novel crowd counting setting, i.e. only using partial annotations in each image as training data. Inspired by the repetitive patterns in the annotated and unannotated regions as well as the ones between them, we design a network with three components to tackle those unannotated regions: i) in an Unannotated Regions Characterization (URC) module, we employ a memory bank to only store the annotated features, which could help the visual features extracted from these annotated regions flow to these unannotated regions; ii) For each image, Feature Distribution Consistency (FDC) regularizes the feature distributions of annotated head and unannotated head regions to be consistent; iii) a Cross-regressor Consistency Regularization (CCR) module is designed to learn the visual features of unannotated regions in a self-supervised style. The experimental results validate the effectiveness of our proposed model under the partial annotation setting for several datasets, such as ShanghaiTech, UCF-CC-50, UCF-QNRF, NWPU-Crowd and JHU-CROWD++. With only 10% annotated regions in each image, our proposed model achieves better performance than the recent methods and baselines under semi-supervised or active learning settings on all datasets. The code is `https://github.com/svip-lab/CrwodCountingPAL`.

1. Introduction

The crowd counting task aims to estimate the total number of persons in static images or dynamic videos. The recent data-driven models have achieved satisfactory results on crowd counting due to the success of CNN [10], but they

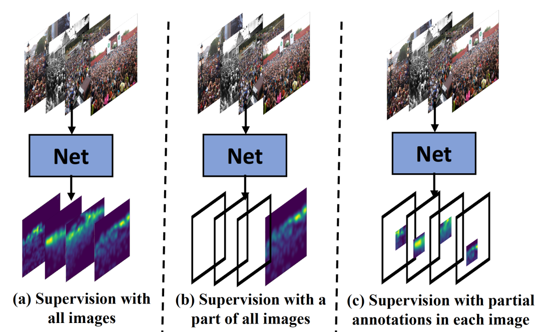


Figure 1. Crowd counting with different supervisions.

still require large amounts of annotated data. For instance, the annotators need to label the positions of all heads with points to overcome the various challenging scenes, such as the lighting, camera view, occlusion and various head poses in Fig. 1 (a). Such a labeling strategy is an extremely labor-intensive task, *e.g.*, the total annotation cost is 3,000 human hours in labeling the NWPU-Crowd dataset [42].

Naturally, a key question arises. Can the designed model still produce the competitive performance but use as few annotations as possible? One of the potential directions is to use a part of the dataset in full annotations, as shown in Fig. 1 (b), under the semi-supervised learning (SSL) [21, 34] or active learning [47] strategy. Although these strategies could reduce the number of annotated training images, we still need to fully annotate the images. It might result in limited challenging scenes, limited viewing angles of the camera as well as limited lighting conditions, which might degrade the model's generalization ability in the test stage.

We notice that in one image, the person's head poses are usually the same or similar and the lighting conditions and the viewing angles are consistent. It might be redundant to annotate all the person heads in one image. Therefore, to fully leverage the data captured from different scenes with different view angles while reducing the annotation cost, we propose a novel crowd counting setting, named **Partial**

*: Equal Contribution. †: Corresponding author.

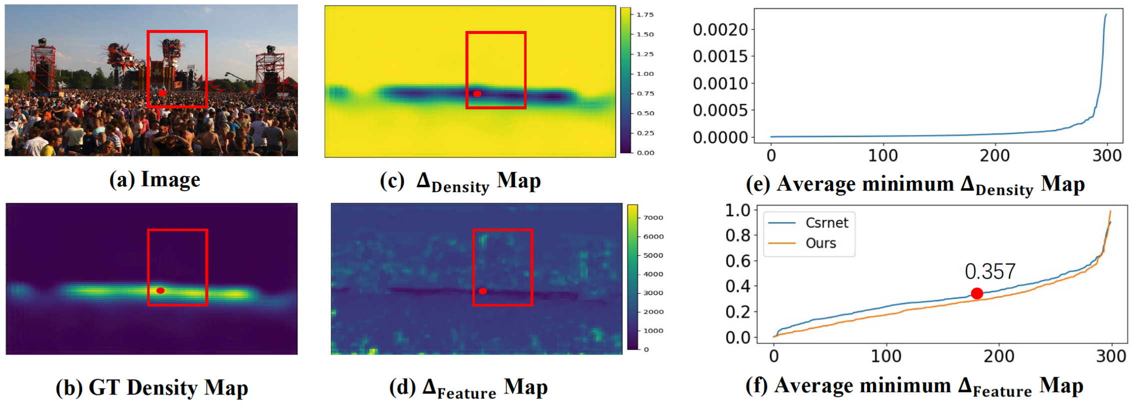


Figure 2. An Illustration of the repetitive pattern. The region within the red bounding box is the partially annotated region. The $\Delta_{Density}$ map shows the density δ between the red point and the rest regions, and the $\Delta_{Feature}$ map represents the feature distance δ between the red point and the rest regions. The blue and orange curves in (f) represent the average minimum distance distribution of the features extracted from the CSRNet (fully-annotated data) and our model (partially-annotated data). The red point is the example used in (a). They indicate that the similar or repetitive patterns (density or feature) occur not only in the annotated regions (within the red bounding box) or unannotated regions (without the red bounding box), but also between the annotated and unannotated regions in almost all images.

Annotation Learning. Different from those attempts that **fully** annotate **a few** training images, our proposed partial annotation learning only **partially** (*e.g.* 10%) annotates a patch in **each** training image. In particular, each image consists of both 10% annotated regions and 90% unannotated regions in Fig. 1 (c). We do it on the whole training images.

One of the main challenges of partial annotation learning is how to leverage the many unannotated regions for learning a good visual representation in each image since the CNN-based methods could extract useful and efficient features for the annotated regions. We observe that the image textures are usually consistent or the same, such as the person's head poses, lighting conditions, and the viewing angles, just like the Fig. 2 (a). Further, to investigate whether there exist such repetitive patterns on feature space, we calculate the distance between the red point and the rest regions in density map space in Fig. 2 (c) and feature space in Fig. 2 (d). At the global level, there is a similar even the same distribution on the deep blue regions, the most crowded region, in both Fig. 2 (c) and (d). Further, we calculate the average minimum distance in each image between each position in the labelled region and the positions in unlabelled region in ground truth density map space in Fig. 2 (e) and feature space in Fig. 2 (f). The blue and orange curves represent the average minimum distance distribution of the features extracted from the CSRNet (fully-annotated data) and our model (partially-annotated data). The red point is the example used in Fig. 2 (a). The almost flat blue curve shows the repetitive feature patterns occur not only in the unannotated or annotated regions but also between the unannotated and annotated regions in almost all images. Thus, it shows the consistency assumption is reliable for almost all images.

Looking back at the main challenge of how to extract the useful features from unannotated regions, we design the following modules at the local and global level to make full use of the repetitive patterns. Firstly, we employ the memory bank idea to store the repetitive feature patterns extracted from the annotated regions in the whole dataset in controlled storage size. Then the features of the unannotated regions could find their nearest counterpart in the memory for the image representation. Consequently, the memory bank could help the information of the annotated regions flow to the unannotated regions. If only considering each feature vector at the local level, it might not be similar or repetitive for the features extracted from annotated and unannotated regions. Besides, since the background consists of unlimited patterns and objects, such as the building or sky, we only consider the consistency of the feature distribution of the person's head regions. Thus, we design a Feature Distribution Consistency regularizer to regularize the features extracted from unannotated head regions have similar feature distribution with those extracted from the annotated head regions. In particular, we firstly forward the network without backward gradient to get the predicted density map as an attention map, which could roughly distinguish the head regions and background regions.

Motivated by previous work [20], we propose to utilize a Cross-regressor Consistency Regularization to learn the visual representations for both annotated and unannotated regions in a self-supervised style. The proposed model includes two branches to estimate density maps generated by Gaussian bandwidth with different sigma. It uses the consistency of crowd numbers between two different estimated density maps within the same image.

The contributions of this work are summarized as follows: (1) To reduce the annotation cost and produce the competitive performance, we propose a novel crowd counting setting, named partial annotation learning, that only annotates a patch of each training image. (2) Inspired by the repetitive patterns, we design an Unannotated Regions Characterization at the local level and Feature Distribution Consistency regularizer at the global level to leverage the unannotated regions for visual representation. (3) Based on the consistency of crowd numbers, we also design a Cross-regressor Consistency Regularization to learn the visual representations in a self-supervised style. (4) The experimental results demonstrate the effectiveness of our proposed model. With only 10% annotated regions in each image, our proposed model achieves better performance than the recent methods and baselines under semi-supervised or active learning settings on all datasets.

2. Related Work

2.1. Crowd Counting

Early crowd counting methods could be roughly divided into detection-based approaches [40, 44, 12] and regression-based approaches [3, 4, 16]. Recently, in view of the success of Convolutional Neural Networks (CNNs) in image classification [37], object detection tasks [6], CNN-based approaches [8, 2, 1, 17, 43, 18, 14, 22, 28, 41, 45] have been widely applied in crowd counting. MCNN [46] is proposed to regress density map for different head sizes with multi-column convolutions. After that, Switch-CNN [30] and CP-CNN [33] are proposed to choose adaptive scales and incorporate contextual information to improve crowd counting. CSRNet [13] is introduced with dilated convolution to expand receptive field. To simultaneously solve counting, density map regression and localization, a composition loss is designed in [9]. Ma *et al.* [23] present a bayesian loss to maximize the predicted expectations of the head with point supervision. Hu *et al.* [7] search for an automatic multi-scale network to extract effective features of heads with Neural Architecture Search (NAS) strategy. To obtain more accurate head localization, some detection-based crowd counting methods [14, 22, 28] and the corresponding networks are also proposed and achieve comparable performance with regression-based methods. Although Wang *et al.* propose a synthetic counting dataset with GTA-V game and a domain adaptation method to alleviate the burden of labeling in real scenarios, there exists still a gap between performances compared with training in real datasets.

2.2. Counting With Limited Label

Manual labeling is very labor-intensive work in crowd counting for those images with dense heads, thus people begin to seek some crowd counting methods with a limited label. An almost unsupervised learning strategy [29] is proposed for dense crowd counting, where almost 99.9% of the parameters of the proposed model are trained without any labeled data. However, the performance of the model is not satisfactory. To obtain a balance between performance and data annotation, Liu *et al.* [20] leverage unlabeled data to rank the number of heads for crowd counting with a self-supervised method. Lei *et al.* [11] design a network that can effectively train models from count-level annotations, which is regarded as a weakly supervised learning. In [21], Liu *et al.* propose a self-training algorithm to incorporate these inter-relationships to generate reliable pseudo-labels for semi-supervised learning. ResNet50-GP [34] is a Gaussian Process-based iterative learning mechanism, using the estimation of pseudo-ground truth for the unlabeled data. Zhao *et al.* [47] propose an active learning framework to gradually label heads.

Since each image includes both the annotated and unannotated regions, these methods [34, 21, 47] cannot be directly applied and need some modifications, such as adding an annotated region mask or cropping these annotated regions and training. Since the annotated regions could be in many small areas, the cropped images might be in low resolution, which might result in the failure of the multi-scale-based methods. More importantly, the semi-supervised learning methods are trained on the limited challenging scenes, which might limit their generalization ability.

Different from works, we introduce a novel partial annotation learning setting for crowd counting, where only requires annotating a small patch in each image. In [15], Lin *et al.* propose a similar block sub-image annotation (50% pixels) as a replacement for full-image annotation. Domain adaptation (DA) mainly tries to align feature distribution gap between inter-images from the different data distribution. Our method also aligns the feature distribution gap between intra-image patches with the same data distribution, where the gaps mainly result from the lack of labels. Some techniques in DA [48] [38], such as adversarial alignment, pseudo label retraining, mean teacher could be further studied to enforce the feature distribution consistency.

3. Method

3.1. Problem Formulation

In this work, we propose a novel partial annotation learning setting for crowd counting. We only annotate one patch of each image. All annotated and unannotated regions are used in the training process.

Given an image $I \in \mathbb{R}^{3\times H\times W}$, the total count number is N, ground truth persons. Under the partial annotation setting, we only annotate a patch $I_{\text{in}} \subset I$, about 10% of $H \times W$ area, and N^p ($N^p < N$) is the number of annotated persons. Then the density map GT of the image I is generated by partial annotations as formulated as:

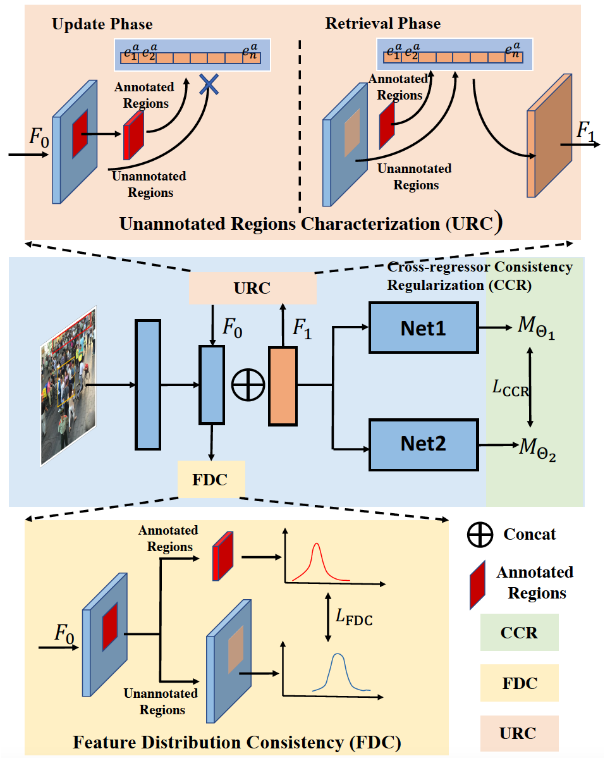


Figure 3. An Illustration of our model.

$GT(p) = \sum_{i=1}^{N^p} \delta(p - p_i) * G_\sigma$, where p_i is the center location of the i^{th} person in the annotated regions.

In our setting, since each image, I includes both the less annotated and more unannotated regions, one of the main challenges is how to extract as much as possible useful visual representation from both of them. Inspired by the repetitive patterns in either annotated or unannotated regions and the ones between them, we propose a model with the three components, as illustrated in Figure 3. In the following, we will describe each module in detail.

3.2. Feature Extraction

Following the CSRNet [13], we use the VGG16 [32] to extract the features and use the same regressor in CSRNet to estimate the density maps. The convolutional layers are used as an encoder and two fully-connected layers are removed. If the size of the input patch is $H \times W$, the feature map F_0 extracted from the Conv5 layer is reduced by $\frac{1}{8}$, via 3 max-pooling layers.

3.3. Unannotated Regions Characterization

Motivated by the repetitive patterns between the annotated and unannotated regions in the feature space, as shown in the Fig. 2 (d), we employ a VQ-based memory to store the repetitive feature patterns from the whole images in controlled storage size. It only stores the visual annotated features instead of the features extracted from the unannotated regions. Consequently, the unannotated features could combine with their nearest counterpart in the memory for better image representation.

In particular, similar to the VQ-VAE [39, 27], there is one memory bank E to encode and store the annotated visual features in the whole dataset. The memory bank E is defined as a latent vector dictionary $E := e_1, e_2, ..., e_n$, where $e_i \in R^{1\times 128}$ denotes the stored feature in the dictionary and n is the total size of the memory.

There are two stages in this part: the update stage and the retrieval stage. To note that, in the update stage, we only use the learned features extracted from these annotated regions, the red patch in Fig. 3, to update the memory back E, whose receptive fields locate or consists of the annotated regions. Specially, the memory bank takes the feature F_0 as inputs. For the feature vector f_j of F_0 in these annotated regions, we find the most similar latent vector e_i in the memory bank, via the L2 distance measure:

$$\hat{f}_j = e_i, i = \arg\min_k \|f_j - e_k\|_2^2. \tag{1}$$

Once the nearest vector e_i is found, we replace the f_j by e_i. Following [39], we use vector quantisation, a dictionary learning algorithm to learn the embedding space. The VQ objective uses l2 error $\|\text{sg}[f] - e\|_2^2$ to move the embedding vectors e_i towards the encoder outputs f_j, which is used for updating the memory E. sg is the stopgradient operator. We use $\|f - \text{sg}[e]\|_2^2$ to make sure that the encoder commits to an embedding and its output does not grow. The vectors in E is learnable from the training set. The latent vectors in memory E are updated only according to these annotated features.

In the retrieval stage, the features extracted from both of the annotated regions, *i.e.*, the red patch in the top of Fig. 3 and unannotated regions, *i.e.*, the blue patches in the top of Fig. 3 need to retrieve the most similar latent vector e_i in the memory bank, similar to Eq. 1. Finally, we concatenate the retrieved features and the original features as the final outputs of the memory bank, donated as F_1.

On the one hand, the unannotated regions could borrow these annotated features via the memory, based on the repetitive patterns between the annotated and unannotated regions. On the other hand, it is easy to expand the trained model to a new domain with the help of the learned annotated features extracted from the whole dataset. To note that, the process of the features extracted from the annotated regions looks like K-means. Both of them can be regarded as a quantization of features. But the K-means method needs to pre-define the number of clusters, while our update phase does not need it.

For learning each feature vector located at the unannotated regions, the simple solution might be to directly retrieve its most similar feature vector from the feature space of the whole image or the same image. However, when retrieving from the whole image, the search space is much large, resulting in a huge time and memory cost. If retrieving from the same image, the search space is limited and lacks enough repetitive patterns. Thus, we employ the memory idea to store the repetitive feature pat-terns extracted from the whole images in controlled storage size.

3.4. Feature Distribution Consistency

A similar feature distribution occurs not only in the annotated or unannotated regions but also between annotated or unannotated regions, as shown in Fig. 2 (d). Since the background consists of all kinds of objects, such as buildings, sky, and so on, we only consider the feature distribution of the person's head regions. These head regions have limited patterns, such as front head or back hair. Inspired it, at the global level, we also design a feature regularizer to regularize the feature distribution extracted from the unannotated head regions similar to one extracted from the annotated head regions as well as the feature distribution within the annotated head regions or unannotated head regions.

In our implementation, we firstly forward the network without backward gradient to get the predicted density map and normalize it as attention map A, which could roughly distinguish the head regions and background regions in both annotated and unannotated regions.

In particular, This module receives the feature F_0 and use as an attention map A. The attention map A is used to filter out the background features both in the non-annotated regions I_{out} and the annotated region I_{in}. The F_{in} and F_{out} represent the features extracted from the annotated patch I_{in} and the unannotated regions I_{out}. Then, we use the mean and covariance of The F_{in} and F_{out} to reduce their differences as follows:

$$\begin{aligned} L_{\text{FDC}} = L_{\text{mean}} + L_{\text{covar}} = \|\mu_{F_{\text{in}}} - \mu_{F_{\text{out}}}\|_2^2 \\ + \|(F_{\text{in}} \cdot A - \mu_{F_{\text{in}}})(F_{\text{in}} \cdot A - \mu_{F_{\text{in}}})^T \\ - (F_{\text{out}} \cdot A - \mu_{F_{\text{out}}})(F_{\text{out}} \cdot A - \mu_{F_{\text{out}}})^T\|_2^2, \end{aligned} \tag{2}$$

where $\mu_{F_{\text{in}}}$ and $\mu_{F_{\text{out}}}$ is the mean vectors of the features extracted from the annotated regions and unannotated regions, respectively. Since the number of the features in annotated regions is different from the number of the features in the unannotated regions, we do not use KL divergence between the two distributions of features.

3.5. Cross-regressor Consistency Regularization

To learn the meaningful visual features of the unannotated regions, we design a Cross-regressor Consistency Regularization (CCR) module in a self-supervised manner. It uses the consistency of crowd numbers between two different estimated density maps.

Given the extracted feature map F_1, we feed it into two branches, *i.e.* Net1 and Net2 to predict density maps generated by the different sigma. All of them use the same front-end network to extract visual features. Here, we denote the predicted density maps from Net1 and Net2 parameterized by θ_1 and θ_2 as $M_{\theta 1}$ and $M_{\theta 2}$, respectively.

Since both Net1 and Net2 use the same images and feature F_1 as inputs, the crowd number of their predicted density maps should be the same. Thus, similar to the self-supervised style in previous work [20] [35], we also use the consistency between their rough predictions as a kind of weakly supervising signals. The loss term is donated as

$$L_{\text{CCR}} = \frac{1}{2N} \sum_{i=1}^{N} \| \sum(M_{\theta 1}^i) - \sum(M_{\theta 2}^i)\|_2^2, \tag{3}$$

where L_{CCR} shows the consistency loss of the density maps predicted by two networks (Net1 and Net2).

To note that, since two branches use two density maps generated by different sigma, the pixel-wise crowd density is different, while the total crowd number should be the same and consistent. Thus, our designed CCR enforces the consistency of the overall crowd number.

3.6. Implementation Details

In our implementation, the final loss function consists of 5 loss items, including two original loss items L_{θ_1} and L_{θ_2}, one cross loss item L_{CCR}, as well as the mean and covariance loss items L_{mean} and L_{covar}. The coefficients of the two original loss items equal 1, while the coefficient of the cross loss item is 0.1. In the FDC module, we use the network prediction as an attention map. Considering the prediction at the beginning phase is not well trained, the coefficient of the mean and covariance loss item increases from 0 to 0.01 during training.

The simulated annotated regions are randomly selected and are rectangle shape. The annotated regions occur at different locations in different images. In the experimental section, we also evaluate the model performance under different annotated shapes such as circles and triangles.

4. Experiment

4.1. Experimental Setting

We use the PyTorch [25] platform to implement our model with the following parameter settings: mini-batch size (16), learning rate (1.0e-6), momentum (0.95), weight decay (0.0005), and the number of epochs (1000). We employ the default initialization to initialize the model.

Dataset. We use the following public datasets to evaluate our proposed model: the ShanghaiTech dataset [46] Part A and Part B, the UCF-CC-50 dataset [9], the UCF-QNRF dataset [9], the NWPU-crowd dataset [42], and The JHU-CROWD++ dataset [35] [36].

Method	Type	Ratio	Part A		Part B	
			MAE	MSE	MAE	MSE
MCNN [46]	FSL	100%	110.2	173.2	26.4	41.3
Switching-CNN [30]	FSL	100%	90.4	135.0	21.6	33.4
CP-CNN [33]	FSL	100%	73.4	106.4	20.1	30.1
ic-CNN [26]	FSL	100%	68.5	116.2	10.7	16.0
PACNN [31]	FSL	100%	62.4	102.0	7.6	11.8
BAYESIAN+ [23]	FSL	100%	62.8	101.8	7.7	12.7
IRAST [21]	SSL	10%	86.9	148.9	14.7	22.9
GP (ResNet-50) [34]	SSL	5%	102	172	15.7	27.9
GP (VGG16) [34]	SSL	5%	112	163	NA	NA
AL-AC [47]	AC	10%	87.9	139.5	12.7	20.4
Label-10% Images	SSL	10%	98.80	165.28	15.88	26.62
Label-10% Regions	PAL	10%	83.87	138.08	16.35	26.11
Ours	PAL	10%	72.79	111.61	12.03	18.70
CSRNet [13]	FSL	100%	68.2	115.0	10.6	16.0

Table 1. The comparison on ShanghaiTech Part A & B dataset.

Metrics. Following the common metrics in existing works for crowd counting, we use both Mean Absolute Error (MAE) and Mean Squared Error (MSE) to evaluate different methods: $MAE = \frac{1}{N}\sum_1^N |z_i - \hat{z}_i|, MSE = \sqrt{\frac{1}{N}\sum_1^N (z_i - \hat{z}_i)^2}$, where N is the number of test images, z_i is the actual number of people in the i^{th} image, and $\hat{z}_i$ is the estimated number of people in the i^{th} image.

4.2. Performance Comparison

We evaluate our proposed model with the following state-of-the-art methods and our designed baselines on the five public datasets, using the metrics MAE and MSE.

Baselines. Since this is the first work to study partial annotation setting in crowd counting, we compare our model with the following methods, divided into three groups.

*Fully-Supervised Learning (**FSL**).* The first group is related to the fully-supervised learning methods. We list some recent state-of-the-art methods for crowd counting, using all samples as training, such as MCNN [46], Switching-CNN [30], CSRNet [13] and so on.

*Semi-Supervised Learning or Active Learning (**SSL/AL**).* We compare our model with the following related semi-supervised or active learning crowd counting methods. IRAST [21] is a self-training algorithm to incorporate these inter-relationships to generate reliable pseudo-labels for semi-supervised learning. ResNet50-GP [34] is a Gaussian Process-based iterative learning mechanism, using the estimation of pseudo-ground truth for the unlabeled data. PSSW [47] is an active learning framework for crowd counting. Besides, we also design a simple baseline, 'Label-10% Images', using the full annotated 10% images to train the network (CSRNet model [13]).

*Partial Annotation Learning (**PAL**).* The third group is under partial annotation learning. We design a simple baseline as low bound, 'Label-10% Regions', using the partial annotation (10% regions) with each image to train the CSRNet [13], using masks on the loss function.

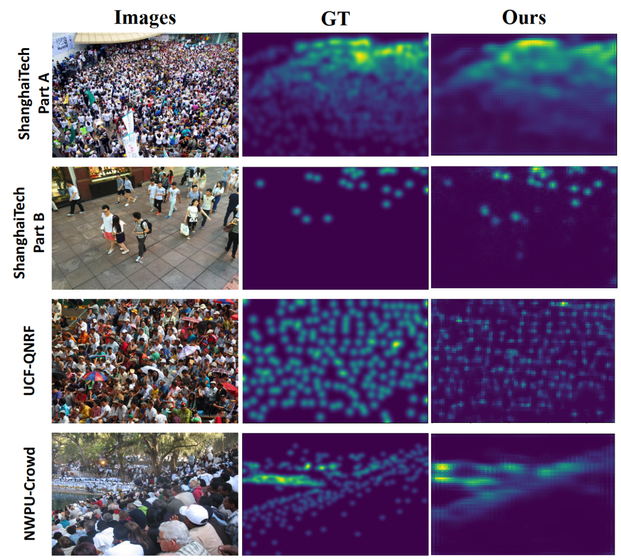


Figure 4. An Illustration of predicted density maps on ShanghaiTech Part A&B, UCF-QNRF and NWPU-Crowd dataset.

Method	Type	Ratio	MAE	MSE
MCNN [46]	FSL	100%	377.6	501.9
Switching CNN [30]	FSL	100%	318.1	439.2
CP-CNN [33]	FSL	100%	295.8	320.9
ic-CNN [26]	FSL	100%	260.0	365.5
PACNN [31]	FSL	100%	241.7	320.7
BAYESIAN+ [23]	FSL	100%	229.3	308.2
AL-AC [47]	AC	20%	318.7	421.6
AL-AC [47]	AC	10%	351.4	448.1
Label-10% Images	SSL	10%	490.11	738.32
Label-10% Regions	PAL	10%	394.75	530.09
Ours	PAL	10%	293.99	443.09
CSRNet [13]	FSL	100%	266.1	397.5

Table 2. The experimental results on the UCF_CC_50 dataset.

The 'Ratio' column on the tables represents how many annotated regions percentage the method uses as training samples. The FSL methods use 100% annotated data. The SSL methods use the 10% fully annotated images as training samples. For the PAL methods, 10% means that each image has a 10% annotated region.

ShanghaiTech Part A & B. The experimental results are shown on the Table 1. We can see that our model achieve a significant performance improvement and is very close to the fully supervised CSRNet, where the gap is less than 2 on the metric of MAE on the ShanghaiTech Part B dataset.

UCF-CC-50. We also conduct experiments on the UCF-CC-50 dataset. On the Table 2, our proposed method even achieves better performance than the early fully supervised counting method such as MCNN [46] and Switching CNN [30]. Besides, our model using only 10% annotations also outperforms the AL-AC using 20% annotations.

Method	Type	Ratio	MAE	MSE
MCNN [46]	FSL	100%	277	426
Switching CNN [30]	FSL	100%	228	445
IRAST [21]	SSL	10%	135.6	233.4
GP (ResNet-50) [34]	SSL	5%	160	275
GP (VGG16) [34]	SSL	5%	175	291
Label-10% Images	PAL	10%	188.33	304.79
Label-10% Regions	PAL	10%	169.04	299.43
Ours	PAL	10%	128.13	218.05
CSRNet [13]	FSL	100%	119.2	211.4

Table 3. The experimental results on the UCF-QNRF dataset.

Method	Type	Ratio	MAE	MSE
MCNN [46]	FSL	100%	218.53	700.61
CANNet [19]	FSL	100%	93.58	489.90
SCAR [5]	FSL	100%	81.57	397.92
SFCN+ [42]	FSL	100%	95.46	608.32
Label-10% Images	SSL	10%	221.94	1172.74
Label-10% Regions	PAL	10%	203.29	1097.55
Ours	PAL	10%	178.70	1080.43
CSRNet [13]	FSL	100%	104.89	433.48

Table 4. The experimental results on the NWPU-Crowd dataset.

Method	Type	Ratio	MAE	MSE
MCNN [46]	FSL	100%	188.9	483.4
SFCN [43]	FSL	100%	77.5	297.6
BCC [24]	FSL	100%	75.0	299.9
DRCN [42]	FSL	100%	71.0	278.6
Label-10% Images	SSL	10%	155.78	463.61
Label-10% Regions	PAL	10%	148.11	409.23
Ours	PAL	10%	129.65	400.47
CSRNet [13]	FSL	100%	85.9	309.2

Table 5. The experimental results on the JHU-CROWD++ dataset.

UCF-QNRF. We then compare our proposed model with other related methods on the UCF-QNRF dataset. Table 3 shows the comparison results. Our proposed method achieves higher performance than other SSL methods even early fully-supervised methods such as MCNN [46] and Switching CNN [30]. Besides, compared with the supervised CSRNet, the gap is less than 10 on MAE and MSE.

NWPU-Crowd and JHU-CROWD++. We also do the comparison on the large scale and widely distributed crowd counting NWPU-Crowd and JHU-CROWD++ datasets. On the Table 4 and 5, our proposed model under the partial annotation setting could achieve better performance than early fully-supervised method MCNN [46]. But since the large scale and various scenes, there still exists a large gap, more than 70 persons in each testing image, between our model and recent fully-supervised method, such as CSRNet [13].

The experimental results on all tables show that our proposed method under partial annotation learning setting always outperforms the recent state-of-the-art semi-supervised learning or active learning methods. Besides, from Fig. 4, we can see our predicted density maps seem like gridding since we use retrieved VQ feature vectors to predict the final density maps.

Method	Branch	Part A MAE	Part A MSE	Part B MAE	Part B MSE
Net1	1	83.87	138.08	16.35	26.11
Net2	1	79.96	122.78	15.46	25.48
Net1&2	2	77.37	119.82	13.51	21.17
Net1&2–URC-CCR-FDC	2	72.79	111.61	12.03	18.70
Net1&2	2	77.37	119.82	13.51	21.17
Net1&2-URC	2	75.27	116.06	12.84	19.74
Net1&2-CCR	2	75.33	119.61	12.60	20.15
Net1&2-FDC	2	75.80	120.26	12.72	19.79
Net1&2–URC-CCR-FDC	2	72.79	111.61	12.03	18.70
Label-Triangle	2	83.06	123.98	14.08	22.23
Label-Circle	2	80.62	120.09	12.88	20.49
Label-Rectangle	2	72.79	111.61	12.03	18.70

Table 6. Ablation studies on the Shanghaitech Part A & B dataset.

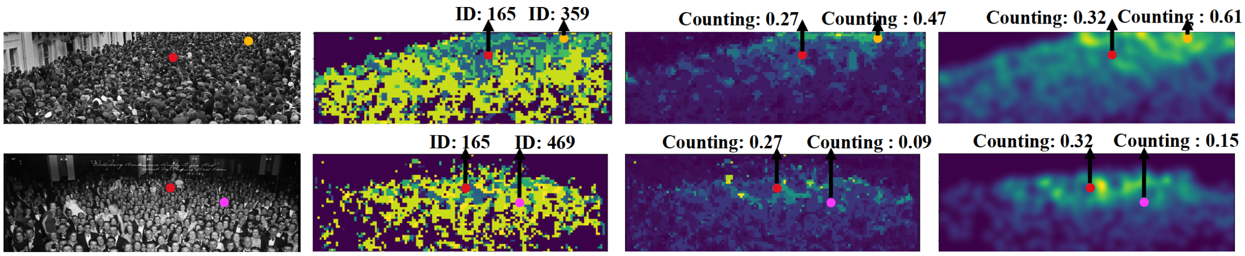
4.3. Ablation Studies

The evaluation of the basic network architectures To evaluate the effects of the basic network architectures, we design the following baselines, 'Net1', 'Net2', and 'Net1&2'. Net1 and Net2 use images with the 10% annotated regions (σ=15 and σ=20) to train the CSRNet[13]. Net1&2 is the multi-branch structure, which shares the encoder and has two or three decoders to predict the density maps. From the first block on Table 6, the multi-branch baselines achieve better performance than the two baselines.

The effect of Unannotated Regions Characterization (URC). In order to evaluate the effect of our proposed URC component, we also design a baseline named 'Net1&2-URC', which adds the UCR component based on 'Net1&2'. Compared with the results of Net-1&2 in the second block on Table 6, Net-1&2-URC achieves fewer errors, which indicates that our proposed URC component could characterize the features of unannotated regions.

The effect of Cross-regressor Consistency Regularization (CCR). To investigate the effect of the Cross-regressor Consistency Regularization (CCR), we train a multi-branch network with a CCR component, named 'Net-1&2-CCR'. Its results are shown in the second block on the Table 6. We can see that it also outperforms the baseline Net-1&2, which indicates that our proposed corss-regressor consistency regularization could learn more useful visual features for the final density maps prediction.

The effect of Feature Distribution Consistency (FDC) To indicate the effect of our proposed feature distribution consistency (FDC), we train a multi-branch network with a CCR component, adding the FDC, named 'Net-1&2-FDC'. The results are shown on the second block on the Table 6. It indicates that FDC could keep alleviates the inconsistency between the annotated and unannotated regions.



(a) Image (b) Indices of VQ vectors (c) VQ density map (d) GT density map

Figure 5. The corresponding relationship between the indices of latent vectors and the crowd counting numbers

The effort of Learned VQ Latent Vector. To investigate the learned latent vectors in the memory bank, we show the corresponding relationship between the indices of latent vectors and the crowd counting numbers, as shown in Fig. 5. After finishing the training process, we feed the whole training images into the trained models and use eq (2) to generate the indices maps of the VQ vectors in Fig. 5 (b). Then we generate the VQ density map based on the correspondence between the indices of the VQ vectors and the counting number. [1] From the Fig . 5 (c) and (d), we can see that there is a high correlation and similar distribution between VQ density maps and ground truth density map. Besides, we can see that even from different images, the similar crowd regions also share the same VQ vector index.

The effect of different shape annotations. We also use different annotated shapes such as circles and triangles to train the model. The third block on Table 6 shows the experimental comparison. On the ShanghaiTech Part B, the gap between circle/triangle shape and rectangle shape is less than 2 in both metrics. However, the gap is much larger about more than 10 persons in each testing image on the ShanghaiTech Part A. The reason might be the receptive field shape used in a neural network, which is more similar to a rectangle shape. For the circle/triangle shape, some boundary elements might be ignored after several convolutional layers, which might result in the performance gap.

The effect of different loss terms. We do the ablation studies on loss terms and the results are in Table 7. We combine the highly related terms: $L_{\theta_1}\&L_{\theta_1}$ and $L_{\text{mean}}\&L_{\text{var}}$.

4.4. Less or More Annotations in An Image

In our implementation, we use all images with 10% annotated regions to train the model. Along with this direction, for each image, we further annotate less regions *i.e.* 5% and more regions *i.e.* 50%, 80% and 90% on the ShanghaiTech Part A and B and use them to train the model. Their point ratio (PR) is ($\frac{\#\text{Points in labelled regions}}{\#\text{All points}}$). The experimental results are shown in the Table 7. The model trained on 5% training data performs worse than the one trained on 10% training data. But the gap is about 7 persons in each testing image on the metric MAE. The model with 90% labels could achieve the similar performance with full supervision, owning to the redundant and repetitive patterns.

[1] The implementation details are in the supplementary materials.

$L_{\theta_1}\&L_{\theta_1}$	L_{CCR}	$L_{\text{mean}}\&L_{\text{var}}$	MAE	MSE	MAE	MSE
1	0.01	0.01	73.85	116.56	12.78	20.23
1	0.1	0.01	72.79	111.61	12.03	18.70
1	1	0.01	136.08	179.01	14.83	23.19
1	0.1	0.1	73.43	113.18	13.10	20.50
1	0.1	0.01	72.79	111.61	12.03	18.70
1	0.1	0.001	73.42	115.15	12.87	20.15
Method	Region Ratio	PR (Part A/B)	MAE	MSE	MAE	MSE
Ours	5%	6.4%/4.0%	79.42	123.60	16.50	25.28
Ours	10%	12.2%/6.6%	72.79	111.61	12.03	18.70
Ours	50%	57.4%/33.3%	70.45	105.03	10.49	16.28
Ours	80%	86.4%/70.0%	67.69	103.71	9.55	14.51
Ours	90%	92.7%/81.2%	67.44	103.75	9.10	13.79
CSRNet	100%	100%/100%	68.20	115.00	10.60	16.00

Table 7. The results with loss weights and more annotations.

Thus, the promising results of the model trained on 5% data indicate there exists further improving space in the crowd counting with partial annotations in an image.

5. Conclusion

To reduce the annotation cost and produce competitive performance, we proposed a novel partial annotation learning setting, only annotating a patch of each image. Compared with semi-supervised learning, our setting could bring more various challenging scenes using the same even less annotation costs. Inspired by the repetitive patterns, we also propose a new model with three modules. With 10% annotated regions in each image, our proposed model always outperforms recent methods under semi-supervised or active learning settings on all datasets. Further, we also train a model using only 5% annotations and the results indicate there is a further improving space under this setting.

Acknowledgments

The work was supported by National Key R&D Program of China (2018AAA0100704), NSFC #61932020, Science and Technology Commission of Shanghai Municipality (Grant No. 20ZR1436000), and 'Shuguang Program' supported by Shanghai Education Development Foundation and Shanghai Municipal Education Commission.

References

[1] Deepak Babu Sam, Neeraj N Sajjan, R Venkatesh Babu, and Mukundhan Srinivasan. Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3618–3626, 2018.

[2] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *The European Conference on Computer Vision (ECCV)*, September 2018.

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

[4] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. Feature mining for localised crowd counting. In *BMVC*, volume 1, page 3, 2012.

[5] Junyu Gao, Qi Wang, and Yuan Yuan. Scar: Spatial-/channel-wise attention regression networks for crowd counting. *Neurocomputing*, 363:1–8, 2019.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[7] 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.

[8] 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 *The European Conference on Computer Vision (ECCV)*, September 2018.

[9] 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 (ECCV)*, pages 532–546, 2018.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[11] Yinjie Lei, Yan Liu, Pingping Zhang, and Lingqiao Liu. Towards using count-level weak supervision for crowd counting. *arXiv preprint arXiv:2003.00164*, 2020.

[12] Min Li, Zhaoxiang Zhang, Kaiqi Huang, and Tieniu Tan. Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *ICPR*, pages 1–4. IEEE, 2008.

[13] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. *CVPR*, pages 1091–1100, 2018.

[14] Dongze Lian, Jing Li, Jia Zheng, Weixin Luo, and Shenghua 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*, pages 1821–1830, 2019.

[15] Hubert Lin, Paul Upchurch, and Kavita Bala. Block annotation: Better image annotation with sub-image decomposition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5290–5300, 2019.

[16] Bo Liu and Nuno Vasconcelos. Bayesian model adaptation for crowd counts. In *ICCV*, pages 4175–4183, 2015.

[17] 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*, pages 1774–1783, 2019.

[18] 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.

[19] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. pages 5099–5108, 2019.

[20] 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.

[21] Yan Liu, Lingqiao Liu, Peng Wang, Pingping Zhang, and Yinjie Lei. Semi-supervised crowd counting via self-training on surrogate tasks. *arXiv preprint arXiv:2007.03207*, 2020.

[22] 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, 2019.

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

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

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

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

[27] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pages 14866–14876, 2019.

[28] Deepak Babu Sam, Skand Vishwanath Peri, Amogh Kamath, R Venkatesh Babu, et al. Locate, size and count: Accurately resolving people in dense crowds via detection. *arXiv preprint arXiv:1906.07538*, 2019.

[29] Deepak Babu Sam, Neeraj N Sajjan, Himanshu Maurya, and R Venkatesh Babu. Almost unsupervised learning for dense crowd counting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8868–8875, 2019.

[30] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. Switching convolutional neural network for crowd counting. In *CVPR*, volume 1, page 6, 2017.

[31] 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.

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[33] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *ICCV*, pages 1879–1888. IEEE, 2017.

[34] Vishwanath A Sindagi, Rajeev Yasarla, Deepak Sam Babu, R Venkatesh Babu, and Vishal M Patel. Learning to count in the crowd from limited labeled data. *arXiv preprint arXiv:2007.03195*, 2020.

[35] Vishwanath A Sindagi, Rajeev Yasarla, and Vishal M Patel. Pushing the frontiers of unconstrained crowd counting: New dataset and benchmark method. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1221–1231, 2019.

[36] Vishwanath A Sindagi, Rajeev Yasarla, and Vishal M Patel. Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. *Technical Report*, 2020.

[37] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[38] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018.

[39] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017.

[40] Paul Viola, Michael J Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. In *null*, page 734. IEEE, 2003.

[41] 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.

[42] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting and localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2020.

[43] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. pages 8198–8207, 2019.

[44] Bo Wu and Ram Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *null*, pages 90–97. IEEE, 2005.

[45] Qi Zhang and Antoni B Chan. 3d crowd counting via multi-view fusion with 3d gaussian kernels. *AAAI*, 2020.

[46] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *CVPR*, pages 589–597, 2016.

[47] Zhen Zhao, Miaojing Shi, Xiaoxiao Zhao, and Li Li. Active crowd counting with limited supervision. *arXiv preprint arXiv:2007.06334*, 2020.

[48] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.