Weakly-Supervised Crowd Counting Learns from Sorting rather than Locations

Yifan Yang[1], Guorong Li[⋆ 1,2], Zhe Wu[1], Li Su[1,2], Qingming Huang[1,2,3], Nicu Sebe[4]

[1]School of Computer Science and Technology, UCAS, Beijing, China
[2]Key Lab of Big Data Mining and Knowledge Management, UCAS, Beijing, China
[3]Key Lab of Intelligent Information Processing, ICT, CAS, Beijing, China
[4]University of Trento, Trento, Italy

Abstract. In crowd counting datasets, the location labels are costly, yet, they are not taken into the evaluation metrics. Besides, existing multi-task approaches employ high-level tasks to improve counting accuracy. This research tendency increases the demand for more annotations. In this paper, we propose a weakly-supervised counting network, which directly regresses the crowd numbers without the location supervision. Moreover, we train the network to count by exploiting the relationship among the images. We propose a soft-label sorting network along with the counting network, which sorts the given images by their crowd numbers. The sorting network drives the shared backbone CNN model to obtain density-sensitive ability explicitly. Therefore, the proposed method improves the counting accuracy by utilizing the information hidden in crowd numbers, rather than learning from extra labels, such as locations and perspectives. We evaluate our proposed method on three crowd counting datasets, and the performance of our method plays favorably against the fully supervised state-of-the-art approaches.

Keywords: Weakly-supervised, Sorting, Multi-frames, Crowd Counting

1 Introduction

Counting objects is a hot topic in computer vision because of its wide applications in many areas. Significant effort has been devoted to this task [34, 15, 18, 38, 30, 2, 21, 11]. These approaches either employ a detection framework [18, 22] or a regression framework [34, 15, 18]. However, in congested scenes, there are many occlusions, and it is difficult for the detection approaches to recognize the person. Therefore, the density estimation based methods [34, 15, 18], in particular, have received increasing research focus.

However, there are still several drawbacks. Firstly, the annotations of the crowd counting are generally expensive. The existing counting datasets [46, 45, 3, 9] provide the location of each instance to train the counting networks, while in the evaluation stage, these location labels are not taken into account, and the performance metrics only evaluate the estimation accuracy of the crowd number.

⋆ Corresponding author. liguorong@ucas.ac.cn

In fact, without the demand for locations, the crowd numbers can be obtained in other economical ways. For instance, with an already collected dataset, the crowd numbers can be obtained by gathering the environmental information, e.g., detection of disturbances in spaces, or estimation of the number of moving crowds. Chan et al. [3] segment the scene by crowd motions and estimate the crowd number by calculating the area of the segmented regions. To collect a novel counting dataset, we can employ sensor technology to obtain the crowd number in constrained scenes, such as mobile crowd sensing technology [10]. Moreover, Sheng et al. [35] propose a GPS-less energy-efficient sensing scheduling to acquire the crowd number more economically. On the other hand, several approaches [17, 4, 14, 23] prove that, in the estimated results, there is no tight bond between the crowd number and the location. Finally, in the existing multi-task approaches, high level tasks are employed to improve the counting accuracy, for instance, tracking [23], detection [18, 22], segmentation [37, 47], localization [25, 17], depth prediction [47] and scene analysis [20, 36, 45, 44, 19]. This research tendency increases the demand for more annotations.

In this work, we propose a weakly-supervised framework to directly regress the crowd number without the supervision of location labels. To our best knowledge, we are the first to train a counting network without location supervision. Moreover, we train the network to count by exploiting the relationship among the images. We propose an end-to-end trainable soft-label sorting network along with the counting network, which sorts the given images by their crowd numbers. The sorting network drives the shared backbone CNN model to obtain density-sensitive ability explicitly. Therefore, the proposed method improves the counting accuracy by utilizing the relationship among crowd numbers, rather than learning from extra labels, such as locations and perspectives. More concretely, the proposed sorting network processes several images and employs a soft-sort layer to generate dense order matrixes. The previous sorting works [16, 8, 6] employ hard-labels to train the sorting network, for instance, one-hot vectors [16, 8] or indexes which are real integers [6]. However, we find that hard-labels are incapable of capturing the complexity of sorting task. As the candidates may have limited variations or even have the same values, the hard-labels introduce ambiguous supervision to the training stage. Therefore, we propose an informative soft-label, which introduces the Rayleigh distribution to characterize the sorting complexity. The proposed soft-labels have high entropy. They provide much more information per training case than hard-labels and much less variance in the gradient between training cases.

The main contributions of our method are summarized as follows:

- We propose a weakly-supervised counting network, which directly regresses the crowd number without the supervision of location labels.
- We propose a soft-label sorting network to facilitate the counting task, which sorts the images by their crowd numbers. The proposed framework improves the counting task without extra labels, especially costly semantic labels.
- The proposed weakly-supervised approach plays favorably against fully supervised state-of-the-art approaches on three datasets.

2 Related Work

2.1 Density Estimation Based Methods

The counting datasets provide a location label for each person. The fully supervised density estimation based methods have to generate density maps with various strategies. Several approaches [46, 15] coarsely estimate the instance scales by the interval distances and employ Gaussian kernels with various scales to represent the objects. Wan et al. [41] propose a network to adaptively generate density maps, and train the generative network with the regression network. The obtained density maps are better recognized by the counting network. However, as proved by several approaches [17, 4, 14, 23], in the estimated density maps, there is no tight bond between the crowd number and the location. In the scenes with large perspective distortions, regardless of the low regression errors, dense-crowd regions are usually underestimated, while sparse-crowd regions are overestimated. Du et al. [23] prove a similar phenomenon in the scenes with limited scale variations.

The existing approaches employ various strategies to improve counting accuracy. Several approaches employ multiple receptive fields to evaluate the instances with various scales. To obtain the multiple receptive fields, Zhang et al. [46], Deb et al. [7] and Sam et al. [29] employ multi-column networks; several approaches [2, 13, 19] utilize inception blocks. Besides changing the convolution kernels, a deep network can also obtain various receptive fields from its different layers. Several counting approaches [34, 2, 20, 13] utilize similar architectures with U-net [28]. However, Li et al. [15] prove that the multiple receptive fields deliver similar results. Moreover, several methods employ extra supervision to improve evaluation accuracy. For instance, Liu et al. [20], Shi et al. [36], Yan et al. [44] and Zhang et al. [45] employ perspective maps to smooth the final density maps. However, the perspective maps are delivered from extra annotations. Besides, the existing multi-task approaches utilize high-level tasks to improve the estimation accuracy, for instance, tracking [23], detection [18, 22], segmentation [37, 47], localization [25, 17], and depth prediction [47]. These multi-task approaches boost the counting task with extra semantic labels.

In this work, we propose a weakly-supervised counting approach, which is trained without location labels. Moreover, we propose a soft-label sorting network to improve the counting accuracy, which sorts the images with various crowd numbers. The proposed framework improves the counting task without extra labels, especially costly semantic labels.

2.2 Methods Dealing with the Lack of Labelled Data

Several approaches are proposed to relieve the expensive labeling work in crowd counting. One of the most relevant works for our method is L2R [21], which facilitates the counting task by ranking the image patches. However, L2R is fully supervised by using the location labels. Moreover, the ranking network only operates on the image patch and one of its sub-patches. Our proposed network is

trained without location labels. Besides, the sorting network processes the whole image, and the number of the candidate images are not fixed.

Wang et al. [42] generate synthetic crowd scenes and simultaneously annotate them. The proposed network is pre-trained on the synthetic dataset and then fine-tuned with real data. Although, this approach improves the counting performance with less expensive labels. Labeling the real data is still expensive and challenging. Loy et al. [24] employ active learning to label more representative frames of the videos. This strategy efficiently releases the laborious work. However, it only works on the video counting task, where the video data are assumed to lie along a low-dimensional manifold. Sam et al. [31] pre-train the feature extractor with several restricted Boltzmann machines progressively in an unsupervised way, but the training of the top regression layers is fully-supervised.

Out of these mentioned approaches, only our method and Loy et al. [24] employ fewer labels to train the networks. Still, only our approach trains the network without the supervision of locations.

2.3 Learning from Sorting

Sorting is used pervasively in machine learning. However, it is also a poor match for the differentiable pipelines of deep learning. Currently, several approaches combine the sorting layers with deep networks. Several works [26, 33, 43] encode the permutations into indexes and train the network to regress the index. While others algorithms [16, 8, 6] propose differentiable operators to directly regress the order.

Several self-supervised approaches employ a sorting task to pre-train the feature extractor. Noroozi et al. [26] first propose a self-supervised network to learn a feature domain by solving Jigsaw puzzles. Inspired by this work, Sermanet et al. [33] propose a similar framework to sort the shuffled video frames, and the learned features are used as video representation. Xu et al. [43] also learn video representations by sorting shuffled frames. However, they employ video clips rather than single frames to train the network. These approaches have several restrictions. Firstly, they employ indexes to represent all the possible permutations and train the network to regress the index. This strategy reduces the information embedded in the supervision labels. For example, in [26], there are 9 elements to sort and the number of possible permutations is 9!=362,880. The massive numbers inhibit the methods to sort more elements; for instance, Xu et al. [43] restrict the number of clips between 2 to 5. Moreover, with the casual encoding strategy, a slight variation between two permutations may cause a dramatic difference in their indexes. Therefore, sorting networks cannot learn a representation efficiently. Otherwise, the feature extractors are pre-trained. Our proposed method employs a dense order matrix to capture the possible permutations and end-to-end trains the sorting network with the counting network.

On the other hand, several approaches propose differentiable soft-sort methods to tackle these issues. Sinkhorn distance [5] has been initially proposed to tackle optimal transportation, while Linderman et al. [16] employ it to address sorting issues. Grover et al. [8] propose an attractive task, which sorts n

numbers between 0 and 9999 given as four concatenated MNIST images. They also propose a differentiable neural sort method to tackle this task. Based on the Sinkhorn method, Cuturi et al. [6] further propose a differentiable soft-sort algorithm, which directly generates the sort and rank indexes of a vector. However, these approaches employ hard-labels to train the network, and this leads to the loss of valuable information.

In this paper, instead of pre-training a feature extractor, we train the sorting network with counting network in an end-to-end manner. In the final layer, we also employ a differentiable soft-sort operator to generate dense order matrixes. Moreover, we propose an informative soft-label to train the sorting network.

3 Method

In this paper, we propose a weakly-supervised counting approach, which does not rely on location supervision for training. Besides, to improve the counting task, we exploit the relationship among images. We propose a novel soft-label sorting network along with the regression network, which sorts the images by their various crowd numbers. Both the regression network and the sorting network share a same backbone, and both networks are end-to-end trained. As both networks estimate the crowd numbers of the given images, they promote each other without extra labels, such as location and perspective.

More concretely, the regression network employs several adaptive pooling layers to formulate a pyramidal feature vector. Besides, the sorting network employs a network to formulate the comparison and uses a differentiable soft-sort layer to generate dense order predictions. Moreover, to train the network effectively, we propose soft-labels, which have high entropy and provide much more information. The soft-labels employ the Rayleigh distribution to characterize the complexity of sorting tasks. We will elaborate on the details of the regression network, the sorting network, and the training method in the following subsections.

3.1 Regression Network

As shown in Fig. 1, the regression network directly regresses the crowd number from the whole frame. Moreover, the front-end of the network, which delivers the pyramidal feature vector, is shared with the sorting network.

In the front-end network, we first employ the first 13 layers of VGG-16 [39] as the backbone of our network, similar to previous methods [1, 32, 40, 15]. The front-end is marked as $G_b(\gamma)$, where γ stands for the parameters, and the output size is 1/8 of the original input size. Then the front-end network regresses a single channel density map based on the extracted features, which is formulated as:

$$f_d = F_d(G_b(x, \gamma), \zeta), \tag{1}$$

where $f_d \in \mathbb{R}^{1WH}$ represents the estimated density map, W, H are the width and height of the feature map respectively. Moreover, $F_d(\cdot)$ denotes the convolution operation, and ζ stands for the parameters of the convolution layers.

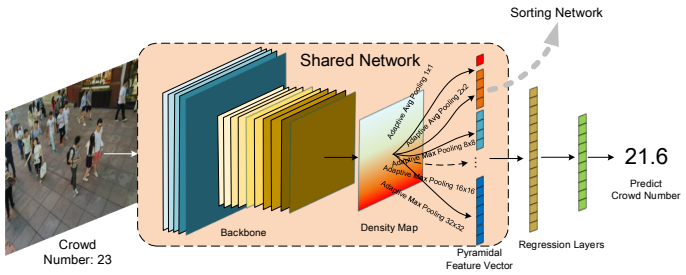

Fig. 1. Framework of the regression network, which contains a shared front-end and a back-end to regress the crowd number.

The network needs the front-end to be sensitive to both the densities of the local and global crowds. Therefore, we propose to use adaptive pooling layers, denoted as $\mathcal{P}$, to extract a pyramidal feature vector from f_d. The adaptive pooling layers consist of global sub-cluster layers and local sub-cluster layers. Each global sub-cluster layer is denoted as $\mathbb{P}_{\mathrm{G}}^{i,S(i)}$, which employs an adaptive average pooling with a high sampling rate $S(i)$ to integrate the global information. Here, $i \in \{1, \cdots, N\}$, N is the number of the global sub-cluster layers. Besides, each local sub-cluster layer is denoted as $\mathbb{P}_{\mathrm{L}}^{j,S(j+N)}$, which employs an adaptive max pooling with a lower sampling rate $S(j+N)$ to extract the most discriminative features. Here, $j \in \{1, \cdots, T-N\}$, T is the total number of the pooling layers. The sampling rates of these pooling layers belong to the sampling rate set, which is denoted as S. We concatenate the outputs of adaptive pooling layers to formulate the pyramidal feature vector:

$$f_{pfv} = \mathcal{P}(f_d, S). \tag{2}$$

The back-end of the regression network employs several fully-connected layers to predict the crowd number:

$$c = F_r(f_{pfv}, \psi), \tag{3}$$

where c is the predicted crowd number, and $F_r(\cdot)$ stands for the regression layers with parameters ψ.

3.2 Sorting Network

To process multi-frames, the sorting network employs a multi-branch network to extract the pyramidal feature vectors. Each branch is shared with the regression network, while all the branches share the same parameters. The multi-branch network is formulated as $G_s(\omega)$, and the pyramidal feature vectors are denoted as $\{f_{pfv}^1, f_{pfv}^2, \cdots, f_{pfv}^K\}$. Here, ω represents the shared parameters, and K is the number of the given images.

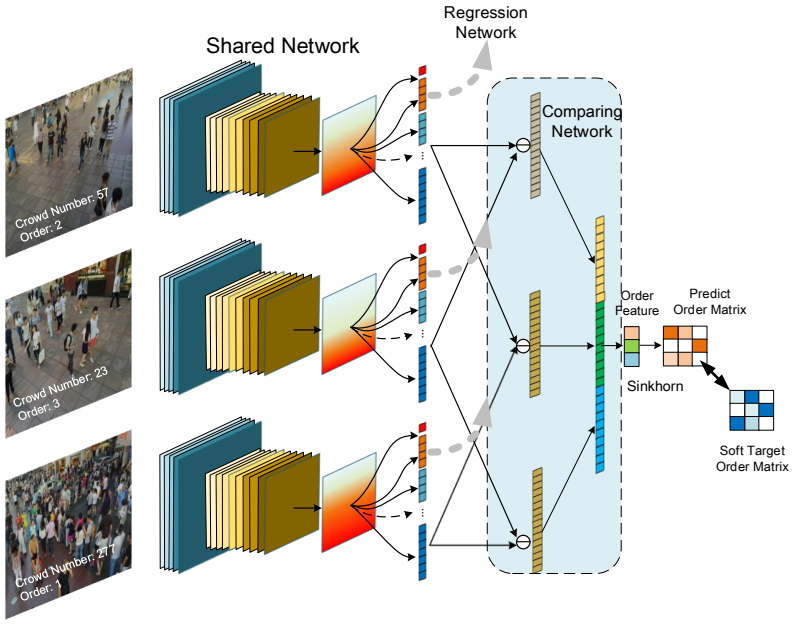

Fig. 2. Framework of the sorting network, which contains several branches to extract the pyramidal feature vectors and a comparing network to regress the order features. The Sinkhorn layer transfers the order feature to an order matrix. We employ the proposed soft-labels to train the sorting network.

The sorting network then utilizes a comparing network to regress the order feature. The comparing network first organizes $K(K-1)/2$ non-repeating tuples, each of which has two elements. Then the network calculates the difference between each pair: $f_{\mathrm{m}}^{ij} = f_{pfv}^{i} - f_{pfv}^{j}$, and concatenates the difference features: $f_{diff} = f_{\mathrm{m}}^{12}||f_{\mathrm{m}}^{13}||\cdots||f_{\mathrm{m}}^{1K}||f_{\mathrm{m}}^{23}||f_{\mathrm{m}}^{24}||\cdots||f_{\mathrm{m}}^{2K}\cdots||f_{\mathrm{m}}^{K-1,K}$, where $||$ denotes the concatenation operation. Finally, the sorting network regresses the order feature of the given images. The order feature is formulated as f_o, where $f_o \in \mathbb{R}^K$.

We employ the Sinkhorn layer [5] to transfer f_o into an order matrix $\mathbf{P}$, where $\mathbf{P}_{ij}$ is the probability that the i-th element is ordered in j, and $\mathbf{P} \in \mathbb{R}^{KK}$. More importantly, we propose a soft-label to characterize the complexity of the sorting task, which is more informative than the hard-label. We elaborate on this in the following subsections.

Sinkhorn Operator The Sinkhorn method is proposed to solve the optimal transportation issue. After several iterations, it generates a matrix to capture the transportation probabilities between two distributions. As the method is differentiable, recently, it has been combined with the deep networks to solve the sorting problems.

In the proposed sorting network, we employ a Sinkhorn layer to generate the transportation matrix between the order feature f_o and the order vector y_o. In the Sinkhorn layer, we first initialize the transportation matrix $\mathbf{P}$ by:

$$\mathbf{P}_{ij} = exp\left(-\frac{|f_o^i - y_o^j|}{\epsilon}\right), \tag{4}$$

where, ϵ is a control factor. In the iterations, the $\mathbf{P}$ is updated as:

$$\mathbf{v} = \frac{1}{\mathbf{P}^\top \mathbf{u} K}, \qquad \mathbf{u} = \frac{1}{\mathbf{P}\mathbf{v} K}, \tag{5}$$

where $\mathbf{u} = \mathbf{1}_K$. The iterations stop when $\Delta(\mathbf{v}\mathbf{P}^\top \mathbf{u}, \mathbf{1}_K/K) < \eta$. The max iteration number l is depend on ϵ: typically, the smaller ϵ, the larger l is needed to ensure that $\mathbf{v}\mathbf{P}^\top \mathbf{u}$ is close to $\mathbf{1}_K/K$.

Soft-Label To train the sorting network, we transfer the permutation σ to the ground truth transportation matrix $\mathbf{P}^{GT}$. Previous works generate a hard label, where $\mathbf{P}^{GT}(i, \sigma(i)) = 1$. However, the sorting task is complex, and the hard-labels are unable to cover all the situations. For instance, there may be several candidates with similar or even identical values. Therefore, we propose soft-labels with high entropy to capture transportation probabilities. They provide not only much more information per training case than hard-labels but also much less variance in the gradient between training cases.

The soft-labels introduce the Rayleigh distribution to capture the relations between one element and its neighbors in the permutation. We denote the differences between one element and its neighbours in permutation as $\Delta_{i+1}, \Delta_{i-1}$, where $\Delta_{i+1} = |c_{\sigma(i)} - c_{\sigma(i)+1}|$. We set a threshold, which is denoted as Δ_{thr}, as the sensitivity of the network. If the difference between the two elements is less than the threshold, the network considers them as being similar instances. The elements of the transportation matrix are calculated as:

$$\widehat{\mathbf{P}}^{GT}(i, \sigma(i) + j) = \frac{(h(j) + \mu)}{\sigma^2} e^{\frac{-(h(j)+\mu)^2}{2\sigma^2}}, j \in \{-1, 0, 1\}. \tag{6}$$

To ensure the correct calculation in the edges, before calculating each element, we pad the matrix, and then crop it after calculations. The rate of both operations is 1.

The $\mu, \sigma, h(x)$ are determined by the differences between neighbours in permutation:

$$\begin{cases} \mu = 1, \sigma = 0.5, h(x) = x; & \Delta_{i+1} > \Delta_{thr}, \Delta_{i-1} > \Delta_{thr}, \\ \mu = 1, \sigma = 1.0, h(x) = x; & \Delta_{i+1} \leqslant \Delta_{thr}, \Delta_{i-1} > \Delta_{thr}, \\ \mu = 1, \sigma = 1.0, h(x) = -x; & \Delta_{i+1} > \Delta_{thr}, \Delta_{i-1} \leqslant \Delta_{thr}, \\ \mu = 2, \sigma = 2.0, h(x) = x; & \Delta_{i+1} \leqslant \Delta_{thr}, \Delta_{i-1} \leqslant \Delta_{thr}. \end{cases} \tag{7}$$

Finally, we obtain the soft-label transportation matrix $\mathbf{P}^{GT}$ by normalizing $\widehat{\mathbf{P}}^{GT}$:

$$\mathbf{P}^{GT}(i, \sigma(j)) = \frac{\widehat{\mathbf{P}}^{GT}(i, \sigma(j))}{\sum_{j=1}^{K} \widehat{\mathbf{P}}^{GT}(i, \sigma(j))} \tag{8}$$

3.3 Training Method

We use a straightforward way to train both the regression network and the sorting network as an end-to-end structure. The first 10 convolutional layers are fine-tuned from a pre-trained VGG-16. For the other layers, the initial values come from a Gaussian initialization with 0.01 standard deviation. Stochastic gradient descent (SGD) is applied with a fixed learning rate.

While training on the image dataset and the video dataset, we employ various sampling strategies. This is because the video surveillance scene pays more attention to the variation of pedestrian flow in a constrained scene. With the image dataset, we randomly select images from the dataset and train the network with their crowd numbers. With the video dataset, we first randomly select the video fragments of the same scene. We then randomly choose images within these clips.

We utilize MSE loss to train the regression network:

$$\mathcal{L}_r = \frac{1}{n}\sum_{i=1}^{n}(c_i^{GT} - c_i)^2, \tag{9}$$

where c_i^{GT} is the ground truth crowd number of i-th image.

We employ the cross-entropy loss to supervise the sorting network:

$$\mathcal{L}_s = -\frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{K}(\mathbf{P}_{ij}^{GT}\ log(\mathbf{P}_{ij}) + (1-\mathbf{P}_{ij}^{GT})\ log(1-\mathbf{P}_{ij})). \tag{10}$$

The total loss is formulated as: $\mathcal{L}_{total} = \mathcal{L}_r + \xi\ \mathcal{L}_s$, where ξ is a scalar to balance the two losses.

4 Experiments

We evaluate our approach on three datasets: WorldExpo10 [12], UCSD [3], and ShanghaiTech [46]. In this section, we first provide the implementation details and evaluation metrics. We then evaluate and compare our method with the previous fully-supervised state-of-the-art approaches [46, 2, 15, 27, 19, 13] on all these datasets. In the last subsection, we present ablation study results on the WorldExpo10 dataset.

4.1 Implementation Details

To avoid over-fitting, we employ dropout layers in the fully-connected layers of both the regression network and sorting network. The ratio of dropout is 0.5. In the regression network, we set the sampling rate set as $S = \{\{1, 2\}_{Avg}, \{8, 16, 32\}_{Max}\}$. As the order feature $f_o \in \mathbb{R}^K$, the sorting network employs different regression networks while sorting the various number of candidates. However, we organize each sorting network with the same structure. In the evaluate and compare subsection, we report the regression results of the proposed

Table 1. The evaluation results on WorldExpo10, UCSD, and ShanghaiTech datasets. SHA represents ShanghaiTech Part A, while SHB represents ShanghaiTech Part B. The results reported on WorldExpo10 are only evaluated with the MAE metric.

	Label		WorldExpo10						UCSD		SHA		SHB	
Method	Location	Crowd Number	Sce.1	Sce.2	Sce.3	Sce.4	Sce.5	Avg.	MAE	MSE	MAE	MSE	MAE	MSE
MCMM [46]	✓	✓	3.4	20.6	12.9	13.0	8.1	11.6	1.07	1.35	110.2	173.2	26.4	41.3
SANet [2]	✓	✓	2.6	13.2	9.0	13.3	3.0	8.2	1.02	1.29	67.0	104.5	8.4	13.6
CSRNet [15]	✓	✓	2.9	11.5	8.6	16.6	3.4	8.6	1.16	1.47	68.2	115.0	10.6	16.0
IG-CNN [27]	✓	✓	2.6	16.1	10.15	20.2	7.6	11.3	-	-	72.5	118.2	13.6	21.1
TEDnet [13]	✓	✓	2.3	10.1	11.3	13.8	2.6	**8.0**	-	-	64.2	109.1	8.2	**12.8**
ADCrowdNet [19]	✓	✓	1.6	15.8	11.0	10.9	3.2	8.5	**0.98**	**1.25**	**63.2**	**98.9**	**8.2**	15.7
Ours	-	✓	3.5	13.2	12.4	13.5	5.4	9.6	1.8	2.8	104.6	145.2	12.3	21.2

method, and the candidate number K is 3. In the ablation study subsection, we report the results of sorting the various number of candidates. In the Sinkhorn layer, we set ϵ as 1e-1, and η as 1e-3. When generating the soft-label, we set Δ_{thr} as 5.0. In the training stage, the learning rate is set to 1e-7, the ξ is set to 1e2, and the batch size is set to 10.

4.2 Evaluation Metrics

Similar to Sam et al. [30], we use the MAE and the MSE for evaluation:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|c_i - c_i^{GT}\right|, \tag{11}$$

$$\text{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|c_i - c_i^{GT}\right|^2}, \tag{12}$$

where N is the number of images in one test set, and $c_i^{GT}$is the ground-truth crowd number.

4.3 Evaluation and Comparison

WorldExpo10 [46] is a video counting dataset. The training set has 3,380 videos in 106 scenes, in which 3,380 frames are labeled with point labels. Besides, the testing set has 5 videos in 5 scenes, and 600 frames are labeled. In each training clip, we randomly select 3 videos with the same scene to ensure that the crowd numbers of chosen images have enough diversity. We list the result comparisons of MAE in Table 1, where our method achieves 9.6 average MAE. Without location supervision, our method plays favorably against the fully-supervised approaches. We visualize the density maps delivered from the internal layer and show a successful example in Fig. 3 (a), where the regression results have low errors. Moreover, the predicted orders are correct, and each prediction has high confidence. When processing the image clip in Fig. 3 (b), although the regression network delivers accurate estimations, the sorting network

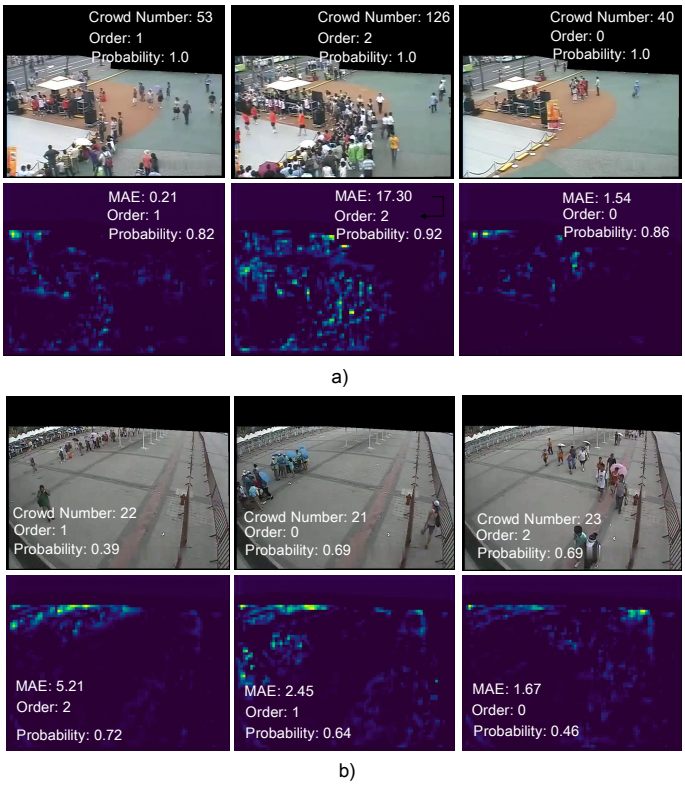

Fig. 3. In the upper row of each example, each image is labeled with its crowd number. Moreover, it is labeled with the order in the tuple and the corresponding probability. In the lower row of each example, we report the MAE of each estimation, and the predicted order with corresponding probability.

encounters a failure. This is because the images have similar crowd numbers. The experiments prove that the proposed counting network can accurately estimate the crowd numbers without location supervision. Moreover, the sorting network is capable of sorting the image clips.

UCSD [3] contains 2,000 frames which are captured by surveillance cameras, and the frames have the same perspective. The comparison between existing approaches and our method is summarized in Table 1. Proved by the results on UCSD and WorldExpo10 datasets, our method overall performs comparably with the fully-supervised approaches in the video surveillance scene.

ShanghaiTech [45] is an image counting dataset. There are 1,198 images with different perspectives and resolutions. This dataset has two parts named Part A and Part B. We report the comparison between our method and state-of-arts in

Table 2. We conduct experiments to verify the efficiency of proposed framework and soft-label on WorldExpo10.

Sorting Network	Regression Network	Soft-Label	MAE	Sorting Accuracy(%)
✓	-	✓	-	50.6
-	✓	✓	20.1	-
✓	✓	-	13.2	**89.1**
✓	✓	✓	**9.6**	78.2

Table 1. Compared with the supervised approaches, our method achieves comparable performance on Part B. While in Part A, our method has a particular gap with other methods. As there is a significant gap between the crowd number distributions of testing set and training set of Part A. More concretely, in the testing set, the mean and standard variance are 354.7 and 433.9, while in the training set, the mean and standard variance are 505.3 and 542.4. On the contrary, the crowd numbers in both sub-sets of Part B have a similar distribution. In the testing set, the mean and standard variance are 95.3 and 124.1, while in the training set, the mean and standard variance are 94.0 and 123.2. The nonlinear regression network can not solve the unbalance distributions of dataset only with the crowd number labels, and needs more powerful supervision, for instance, the location and perspective labels.

4.4 Ablation Study

In this section, we conduct several experiments to study the effect of different aspects of our method on WorldExpo10 and show the results in Table 2 and Table 3. In the first part, we conduct experiments to verify the dependency between the sorting and regression tasks. In the second part, we experiment to verify the efficiency of the proposed soft-label. While in the last part, we test several modifications to the proposed method.

Only Sorting and Only Regression To verify the dependency between the sorting and regression networks, we train the two networks separately and report the results in Table 2. When we train the sorting network alone, the sorting accuracy decreases by 35.3%. While we train the regression network alone, the regression accuracy decreases by 109.4%. The experiments demonstrate that the two tasks can promote each other. This is because the two tasks both estimate the crowd numbers and are closely related.

Soft-Label and Hard-Label We employ the hard-label to train the proposed framework and report the results in Table 2. Each line of the hard-label is a one-hot vector. The sorting accuracy obtains 13.4% improvement, while the performance of counting task drops by 28.1%. This is because soft-labels have high entropy, and it is hard for the sorting network to predict accurate transportation probabilities. However, the soft-labels also contain much more information. Thus, they facilitate the counting task to improve performance.

Table 3. We evaluate several modifications to the proposed method, and report the results on WorldExpo10.

Backbone	Pooling Cluster	Frame Number	MAE	Sorting Accuracy(%)
R3D	MAx & Avg	3	10.1	72.2
VGG	Avg	3	11.4	58.5
VGG	MAx & Avg	4	12.5	30.1
VGG	MAx & Avg	5	13.1	16.3
VGG	MAx & Avg	3	**9.6**	78.2

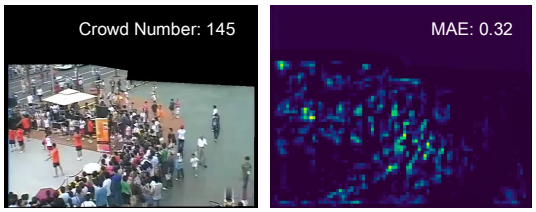

Fig. 4. In the left image, we show an example frame and label its crowd number. While in the right image, we show the corresponding density map, which is an internal feature map of the network. Moreover, we label the density map with the estimation error.

Different Backbones When evaluating on the video datasets, we employ images and clips to train the network, respectively. Each video clip contains 5 frames, which are sampled every 10 frames. To process the video clips, we employ the R3D network [43] as the backbone, which uses 3D convolutional layers and residual connections. The R3D network obtains improvements on several tasks, for instance, action recognition. However, as shown in Table 3, the performance of the modified method drops by 5.2%. The experiments show that the time dimension has a limited influence on the counting results. In the counting dataset, there is no regular pattern for the crowd movement. Therefore, the 3D convolutional layers extract less discriminative features.

Different Frames Numbers We employ various numbers of frames to train the sorting network and report the results in Table 3. As mentioned above, the regression networks of the candidate sorting networks have various structures. However, regression networks have the same structure. The sorting task is more difficult while sorting more candidates. For instance, when sorting 3 images, a randomly guess has a probability of 1/6 to be right, while sorting 5 images, the right probability drops to 1/120. Therefore, the sorting accuracies of 4-candidates network and 5-candidates decrease by 61.5% and 79.1%, respectively. The regression accuracies of the two networks drop by 30.2% and 36.4%. This phenomenon affirms that more accurate sorting facilitates the counting task more. However, the regression accuracies of both candidates are still higher than that of the one without the assistant of sorting network.

Different Sampling Methods In the counting network, we employ various pooling cluster operations to extract the features. The candidate method employs the same sampling rates, yet all the layers use adaptive average pooling layers. The performance of the modified network drops by 18.8%. This result suggests that max-pooling layers are more efficient while extracting the local features. In Fig. 4, we show an example, which is an internal density map delivered from the proposed method. As the density map is noisy, the max-pooling layers extract most discriminative features. Meanwhile, the internal feature map is not supervised by the artificial density maps. Therefore, the responses of this density map are not ideal Gaussian signals. The obtained density map maintains the original semantic information. This phenomenon affirms that without the demand for regressing the instance locations with Gaussian kernels, the counting network concentrates on regressing the crowd number. Therefore, the weakly-supervised crowd counting is a promising research tendency.

5 Conclusions

In this paper, we propose a weakly-supervised counting method, which is trained without location supervision. Moreover, we exploit the relationship among the images to improve the counting accuracy. We propose a novel soft-label sorting network along with the counting network, which sorts the given images by their crowd numbers. We train end-to-end both the sorting network and the regression network. During training, the sorting network drives the shared backbone CNN model to obtain density-sensitive ability explicitly. Therefore, the proposed method improves the counting accuracy by using the information among crowd numbers, rather than learning from extra labels. In the proposed sorting network, we propose a more informative soft-label to capture the complexity of the sorting task. We conduct experiments on three datasets and compare the proposed weakly-supervised approach with the fully-supervised methods. Extensive experimental results demonstrate the state-of-the-art performance of our method. In future work, we will propose a corresponding weakly-supervised benchmark to facilitate this task.

Acknowledgements

This work was supported in part by the Italy-China collaboration project TALENT:2018YFE0118400, in part by National Natural Science Foundation of China: 61620106009, 61772494, 61931008, U1636214, 61836002 and 61976069, in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013, in part by Youth Innovation Promotion Association CAS.

References

1. Boominathan, L., Kruthiventi, S.S.S., Babu, R.V.: Crowdnet: A deep convolutional network for dense crowd counting. ACM Multimedia pp. 640–644 (2016)

2. Cao, X., Wang, Z., Zhao, Y., Su, F.: Scale aggregation network for accurate and efficient crowd counting. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 757–773 (2018)
3. Chan, A.B., Liang, Z.S.J., Vasconcelos, N.: Privacy preserving crowd monitoring: Counting people without people models or tracking. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1–7 (2008)
4. Cheng, Z., Li, J., Dai, Q., Wu, X., Hauptmann, A.G.: Learning spatial awareness to improve crowd counting. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 6152–6161 (2019)
5. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transportation distances. Neural Information Processing Systems pp. 2292–2300 (2013)
6. Cuturi, M., Teboul, O., Vert, J.: Differentiable ranks and sorting using optimal transport. Conference on Neural Information Processing Systems (2019)
7. Deb, D., Ventura, J.: An aggregated multicolumn dilated convolution network for perspective-free counting. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 195–204 (2018)
8. Grover, A., Wang, E.H., Zweig, A., Ermon, S.: Stochastic optimization of sorting networks via continuous relaxations. International Conference on Learning Representations (2019)
9. Guerrerogomezolmedo, R., Torrejimenez, B., Lopezsastre, R.J., Maldonadobascon, S., Onororubio, D.: Extremely overlapping vehicle counting. Iberian Conference on Pattern Recognition and Image Analysis. pp. 423–431 (2015)
10. Guo, B., Wang, Z., Yu, Z., Wang, Y., Yen, N.Y., Huang, R., Zhou, X.: Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. ACM Computing Surveys **48**(1), 7:1–7:31 (2015)
11. Huang, S., Li, X., Zhang, Z., Wu, F., Gao, S., Ji, R., Han, J.: Body structure aware deep crowd counting. IEEE Transactions on Image Processing **27**(3), 1049–1059 (2018)
12. Idrees, H., Saleemi, I., Seibert, C., Shah, M.: Multi-source multi-scale counting in extremely dense crowd images. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2547–2554 (2013)
13. Jiang, X., Xiao, Z., Zhang, B., Zhen, X., Cao, X., Doermann, D., Shao, L.: Crowd counting and density estimation by trellis encoder-decoder network. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
14. Lempitsky, V.S., Zisserman, A.: Learning to count objects in images. In: NIPS (2010)
15. Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1091–1100 (2018)
16. Linderman, S.W., Mena, G., Cooper, H., Paninski, L., Cunningham, J.P.: Reparameterizing the birkhoff polytope for variational permutation inference. International Conference on Artificial Intelligence and Statistics. (2017)
17. Liu, C., Wen, X., Mu, Y.: Recurrent attentive zooming for joint crowd counting and precise localization. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
18. Liu, J., Gao, C., Meng, D., Hauptmann, A.G.: Decidenet: Counting varying density crowds through attention guided detection and density estimation. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5197–5206 (2018)

19. Liu, N., Long, Y., Zou, C., Niu, Q., Pan, L., Wu, H.: Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
20. Liu, W., Salzmann, M., Fua, P.: Context-aware crowd counting. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
21. Liu, X., De Weijer, J.V., Bagdanov, A.D.: Leveraging unlabeled data for crowd counting by learning to rank. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 7661–7669 (2018)
22. Liu, Y., Shi, M., Zhao, Q., Wang, X.: Point in, box out: Beyond counting persons in crowds. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
23. Longyin, W., Dawei, D., Pengfei, Z., Qinghua, H., Qilong, W., Liefeng, B., Siwei, L.: Drone-based joint density map estimation, localization and tracking with space-time multi-scale attention network. arxiv (2020)
24. Loy, C.C., Gong, S., Xiang, T.: From semi-supervised to transfer counting of crowds. International Conference on Computer Vision pp. 2256–2263 (2013)
25. Ma, Z., Wei, X., Hong, X., Gong, Y.: Bayesian loss for crowd count estimation with point supervision. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
26. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. European Conference on Computer Vision pp. 69–84 (2016)
27. Ranjan, V., Le, H., Hoai, M.: Iterative crowd counting. European Conference on Computer Vision pp. 278–293 (2018)
28. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer Assisted Intervention pp. 234–241 (2015)
29. Sam, D.B., Babu, R.V.: Top-down feedback for crowd counting convolutional neural network. National Conference on Artificial Intelligence pp. 7323–7330 (2018)
30. Sam, D.B., Sajjan, N., Babu, R.V., Srinivasan, M.: Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3618–3626 (2018)
31. Sam, D.B., Sajjan, N.N., Maurya, H., Radhakrishnan, V.B.: Almost unsupervised learning for dense crowd counting. Association for Advancement of Artificial Intelligence **33**, 8868–8875 (2019)
32. Sam, D.B., Surya, S., Babu, R.V.: Switching convolutional neural network for crowd counting. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4031–4039 (2017)
33. Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S.: Time-contrastive networks: Self-supervised learning from video. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
34. Shen, Z., Xu, Y., Ni, B., Wang, M., Hu, J., Yang, X.: Crowd counting via adversarial cross-scale consistency pursuit. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
35. Sheng, X., Tang, J., Xiao, X., Xue, G.: Leveraging gps-less sensing scheduling for green mobile crowd sensing. IEEE Internet of Things Journal **1**(4), 328–336 (2014)
36. Shi, M., Yang, Z., Xu, C., Chen, Q.: Revisiting perspective information for efficient crowd counting. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
37. Shi, Z., Mettes, P., Snoek, C.G.M.: Counting with focus for free. International Conference on Computer Vision pp. 4200–4209 (2019)

38. Shi, Z., Zhang, L., Liu, Y., Cao, X., Ye, Y., Cheng, M., Zheng, G.: Crowd counting with deep negative correlation learning. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5382–5390 (2018)
39. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (2015)
40. Sindagi, V.A., Patel, V.M.: Generating high-quality crowd density maps using contextual pyramid cnns. International Conference on Computer Vision pp. 1879–1888 (2017)
41. Wan, J., Chan, A.B.: Adaptive density map generation for crowd counting. International Conference on Computer Vision pp. 1130–1139 (2019)
42. Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from synthetic data for crowd counting in the wild. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
43. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-supervised spatiotemporal learning via video clip order prediction. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 10334–10343 (2019)
44. Yan, Z., Yuan, Y., Zuo, W., Tan, X., Wang, Y., Wen, S., Ding, E.: Perspective-guided convolution networks for crowd counting. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
45. Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 833–841 (2015)
46. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 589–597 (2016)
47. Zhao, M., Zhang, J., Zhang, C., Zhang, W.: Leveraging heterogeneous auxiliary tasks to assist crowd counting. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 12736–12745 (2019)