Scale-Informed Density Estimation for Dense Crowd Counting

Zirui Li [1], Yuexian Zou [1,3,*], Guoshuai Wang [1], Jian Zhang [2,3]
[1] *ADSPLAB, School of ECE, Peking University, Shenzhen, China*
[2] *School of Electrical and Data Engineering, University of Technology Sydney, Sydney, Australia*
[3] *Peng Cheng Laboratory, Shenzhen, China*
*Corresponding author: zouyx@pku.edu.cn

***Abstract* — Dense crowd counting (DCC) remains challenging due to the scale variation and occlusion. Several deep learning based DCC methods have achieved the state-of-arts on public datasets. However, experimental results show that the scale variation is still the main factor to hinder the DCC performance. In this work, we propose a scale-informed dense crowd counting method focusing on handling the negative effect caused by scale variation. More specifically, we propose a method to obtain the scale information of the patch from its GT density maps via estimating the mean value of the Gaussian kernel width and then a scale-classifier is deigned and trained accordingly. Moreover, with the estimated scale information, two sub-nets are dedicatedly deigned to learn the density maps for large-scale head patch and small-scale patch separately. Experimental results validate the performance of our proposed method which achieves the best performance on three dense crowd datasets.**

Keywords—Crowd Counting, Density Estimation, Classifier, Gaussian Kernel Width, Convolutional Neural Network

I. INTRODUCTION

Crowd counting has drawn a great deal of interest among researchers over the last few years. And dense crowd counting (DCC) is a meaningful but thorny sub-problem of crowd counting. On the one hand, dense crowd counting assisted flow monitoring is one of the important applications of public security; On the other hand, the performance of DCC still has room for improvement. Analyzing shows that the large scale variation and severe occlusion are two main factors hindering the performance of DCC methods.

Traditional crowd counting (CC) methods generally fall into two classes: counting by detection [1, 2] and counting by regression [3]. Experiments show that the performance of these methods degrades when there are severe occlusion and serious scale variation. Besides, these tradition CC methods usually ignore the spatial distribution information in crowds. To overcome these problems, an influential CC method has been proposed in [4] where the crowd counting task has been converted to a density map estimation whose integral is the count of objects in the image. As the result, this influential CC method is termed as the density estimation-based counting method.

Recently, deep learning based density estimation counting methods become mainstream [5-6] where the CNN is adopted to learn a mapping from the images to the corresponding density maps. It can be seen that the CC methods have achieved great success in sparse crowd counting [5-9], however, most of them struggle in counting people in the dense scene. According to the experimental results, the DCC method in [10] has achieved state-of-the-art on the dense crowd dataset, like UCF_CC_50 [11] which contains about 1280 people per image. Taking a close look at the experiments

(a) (b)

Fig. 1. (a) a crowd image with large scale variation; (b) the density map of (a). The brightest region in (b) asociated with the high density value (small scale head region). The dim region reflects the low density value (large scale head region). (a) is divided into 4 patches by orange solid line.

in [10], we find that there are still 260.9 in mean absolute error, which indirectly indicates that the dense crowd counting is still required further investigation.

As shown in Fig. 1(a), in the dense scene, the scale variation caused by camera angle is one of the main factors causing the performance degradation. Here, we give some discussion on how does scale variation affect the accuracy of dense crowd counting. In Fig. 1, one dense crowd scene image and its ground-truth (GT) density map are given in (a) and (b) respectively. Comparing (a) and (b), we can see that the large-scale human heads correspond to low density value, and the small-scale heads correspond to high density value. This phenomenon is caused by the GT density map generation, where each head is blurred by a varied Gaussian kernel. This density map generation with dotted annotation is great with a natural property that it models the spatial distribution of crowd. However, it brings difficulties in accurate counting. As shown in Fig. 1, the head scale varies heavily. In the model training, it is probably that the information provided by the small heads may gain more attention while those provided by the big heads might be suppressed. From Fig. 1, it is also noted that the scale variation of the image-level is much larger than the one of the patch-level. For example, the scale variation of a patch (left upper patch in Fig. 1) is much smaller than the scale variation of the whole image.

According to discussions above, in this study, we aim at improving the performance of dense crowd counting by focusing on reducing the adverse effect of scale variation by designing a new scale-informed multi-channel CNN network. First, a scale label is obtained to represent the average size of human heads in one patch (4 patches shown in Fig. 1(a)). Here, the width of Gaussian kernel (sigma) is estimated from the GT density maps to determine the scale label of the patch. The estimated scale labels are used to train a scale-classifier. The details are presented in Section II B. Secondly, two sub-nets are dedicatedly designed to estimate the density maps for two different scale levels (large scale and small scale) where the scale-classifier essentially is used as an adaptive switch to select the sub-net for the input patch as shown in Fig.2.

978-1-7281-3723-0/19/$31.00 ©2019 IEEE

The main contributions of this paper lie in: 1) To decrease the adverse effect of scale variation, a scale-informed dense crowd counting network is designed. 2) A scale-classifier is proposed to infer the scale labels of patches. 3) two subnets are delicately designed for accurate density regression. 4) Intensive experiments have been conducted to evaluate the effectiveness of our proposed method on three datasets.

II. Proposed Method

A. Principle

In this study, we follow the baseline of density estimation method proposed in [4], where the density map is estimated and the count is obtained by the integral of the density map. The generation of the GT density map is introduced and some discussions are given in the following.

The generation of the GT density maps is a key for the density based counting methods. Following the approach proposed in [4], the heads in the training images will be dot annotated and then each annotated head is blurred by convolving with a normalized 2D Gaussian kernel. In [4], single Gaussian kernel is used where the variance of Gaussian is preset. To obtain more reasonable estimation of the GT density maps for dense scenes, the geometry-adaptive kernels (GAK) method was proposed by Zhang et al. [6]. In principle, GAK is much more suitable for dense crowd counting and now is widely used for dense crowd counting. According to [6], the density map of annotated image is generated by (1):

$$\mathrm{F}(p)=\sum_{i=1}^{M}\delta(p-p_i)\otimes G_{\sigma_i}(p) \qquad (1)$$

$$\text{with } \sigma_i=\beta\overline{d^i} \text{ and } \overline{d^i}=\left(\sum_{j=1}^{m}d_j^i\right)/m$$

In (1), i is the annotation dot index and $\otimes$ is the convolution. For each annotated heads p_i, using $\delta(p\text{-}p_i)$ to convolve with a 2D Gaussian kernel G_{σ_i}, σ_i termed as sigma, is the variance of Gaussian kernel which is decided by the mean distance $\overline{d^i}$ between p_i and its m nearest neighborhoods. β is a scaling factor. In this study, following [6], we set $m=3$ and $\beta=3$.

Essentially, the annotated dots of the large-scale heads are much spatially scattered than those of the small-scale heads. Hence, from (1), σ_i is proportional to the spatial distribution of the heads. For the large-scale head, σ_i is much larger than that of the small-scale head. Moreover, from the property of 2D Gaussian kernel function, larger σ_i leads to flatter Gaussian kernel and smaller σ_i leads to a sharper Gaussian function. Accordingly, at annotated dot, the large-scale head might have a small-density value, and small-scale head may have a high-density value at annotated dot.

With the analysis above, for crowd counting task, the GT density map generated by GAK in (1) are dependent on the scale of the heads in the image. Naturally, the mean value of σ_i provides the scale information of the heads in the images.

In this study, to reduce the adverse effect of the scale variation on the estimation of the density maps, we have two intrinsic ideas. First, each image in the training set is divided into 4 patches as shown in Fig.1 (a). Then, all patches are classified into two scales accordingly, they are large-scale patches and small-scale patches, respectively. Second, two subnets (SS-module and LS-module as shown in Fig. 2) are designed to learn density maps of large-scale or small-scale patches separately. To achieve this, three training sets have been formed. The details are given in Section III.

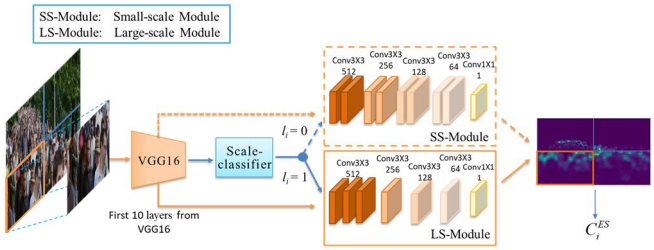

Fig. 2. The architecture of our proposed network. The input patch is fed into VGG16, the scale-classifier predicts the scale label l_i (l_i =0 for small-scale and l_i =1 for large-scale). SS-module or LS-module are selected according to l_i to estimate the density map patch. The count number is the sum over the density map patches. For notation simplicity, VGG16, scale classifier with SS-module or LS-module is named as SS-Net and LS-Net, respectively.

Our goal is to design a scale-informed end-to-end density map learning network for improving counting accuracy for dense scenes. Our main work is presented as follows.

B. Proposed Network

1) Network configuration

The architecture of our proposed network is shown in Fig.2. The network consists of a front-end network, a scale-classifier and two subnets. To be specific, the first 10 layers from the pre-trained VGG16 [12] are employed as our backbone network to extract visual features. Using the scale labels of the patches, a scale-classifier is designed and trained to classify the patches into two categories, small-scale patch or large-scale patch. Two sub-nets, named as SS-module and LS-module, are designed. SS-module maps the features of the small-scale patches to their density maps while SL-module does the same work for large-scale patches. In the prediction stage, the integral of the estimated density map generates the estimated count.

2) Scale-Classifier

Our scale-classifier is a simple network which consists of two fully connected layers with 512 and 2 neurons respectively. The training pairs are input image patches and their associated scale labels.

The generation of scale labels is proposed as follows. First, binary GT density maps are generated using the annotation dot maps where the pixel of annotated dot is signed as 1 and the rest is 0. As a byproduct of generating GT density map, the σ_i at each annotation dot is computed according to (1), then a sigma map is generated using σ_i accordingly. Therefore, the sigma map and annotation dot map have the same size but have different values on each annotated dot.

With the sigma maps of the patches, the mean value of the sigma of a patch can be computed which is a good scale indicator. If the mean value of the sigma of the i-th patch is less than the threshold, the patch scale label l_i is set as 0 which indicates that the i-th patch is with small-scale heads, otherwise, l_i is set to 1 for the patch with large-scale heads. Therefore, the scale label for i-th patch is computed as follows.

$$l_i=\begin{cases}0 & \text{if } \overline{\sigma^i}\le\overline{\sigma}_{th}\\ 1 & \text{if } \overline{\sigma^i}>\overline{\sigma}_{th}\end{cases} \qquad (2)$$

where i is the patch index, $\overline{\sigma}_{th}$ is the threshold and $\overline{\sigma^i}$ is the mean value of sigma which is computed in (3), respectively.

$$\overline{\sigma^i}=\left(\sum_{j=1}^{m}\sigma_j^i\right)/m \quad \text{and} \quad \overline{\sigma}_{th}=\sum_{i=1}^{n}\overline{\sigma^i}/n \qquad (3)$$

where n is the number of patches, σ_j^i is the sigma value of j-th annotated dot in i-th patch.

For training the scale-classifier, the cross-entropy is taken as the loss function which is given in Section II C.

3) SS-module & LS- module

As mentioned in Section II A, the density value is related to the scale of the head. From experiments, we observe that the density values generated using (1) for large-scale heads and small scale heads varies a lot. Bearing these observation in mind, in this study, corresponding to two sub-datasets (small-scale and large-scale patches), two CNN-based sub-networks are designed. In details, considering the filters with different receptive fields bring little effect in counting [8], and in order to limit the net parameters, in our design, only 3×3 convolution kernels are used. As the density map estimation is sensitive to the spatial distribution of the heads, inspired by Li et al. [8], the dilated convolution is used to enlarge receptive fields without losing resolution. In our sub-nets, we set dilation rate as 2. After each convolutional layer, ReLU is employed as activation function.

C. Loss Function

Following the most work, the Euclidean distance is taken as loss function for density estimation task given as follows:

$$L_{ED} = \left(\sum_{i=1}^{N} (F(X_i;\Theta) - D_{GT_i})^2 \right) / (2N) \quad (4)$$

where $F(X_i;\Theta)$ denotes the *i*-th density map generated by our network with parameters Θ . N is the number of training patches, X_i is the *i*-th patch and D_{GT_i} represents the GT density map of the *i*-th patch.

In addition, the cross-entropy is taken as cost function for training the scale-classifier, which is formulated as:

$$L_C = -\left[l_i \log(l_i^{ET}) + (1-l_i)(1-\log(l_i^{ET})) \right] / N \quad (5)$$

where l_i is the scale label of *i*-th patch, which is defined in (3), and l_i^{ET} is the output of the scale-classifier.

Our network is jointly trained using L_{JT} given as:

$$L_{JT} = L_{ED} + \lambda L_C \quad (6)$$

where λ is the predefined weighting factor.

III. Experiments and Results

A. Evaluation Metrics

Two widely used metrics are taken for performance evaluation which are given as follows:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|C_i^{ES} - C_i^{GT}\right| \quad \text{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(C_i^{ES} - C_i^{GT}\right)^2} \quad (7)$$

where MAE is the mean absolute error and MSE is the mean square error. C_i^{ES} is the estimated count of the *i*-th image and C_i^{GT} is the ground-truth count of the *i*-th testing image. N is the number of testing samples. C_i^{ES} and C_i^{GT} are obtained by the integral of estimated and GT density maps respectively.

B. Experiment Details

Except VGG16, normal distribution (std=0.01) is used to initialize our network and all layers are trained from scratch. For training images, each image is divided into four non-overlapping patches (Fig. 1 (a)). For UCF_CC_50 dataset, to increase the training data, we randomly flip all the patches.

For notation clarity, let's define three training sets: 1) Large-scale patch dataset: $T_L = (I_L, D_L, l_i = 1)$; 2) Small-scale patch dataset: $T_S = (I_S, D_S, l_i = 0)$; 3) Whole patch dataset T_A : the combined set of T_L and T_S. Here, I_L and I_S represent the image patches while D_L and D_S represent their density maps.

Our network shown in Fig. 2 is trained in three steps: First, using (6), the LS-Net is trained on T_L. Second, SS-Net is trained on T_S. Third, the LS-Net and SS-Net are jointly trained on T_A. Note that in the joint training step, the classifier is trained to predict the correct scale label information.

Besides, in our network, there are three max pooling layers , the output spatial resolution is then down sampled to 1/8 of original image. Accordingly, the ground truth density map is down-sampled to 1/8. Our network is implemented using Pytorch. In training process, the SGD with momentum is used as the optimization method where the learning rate and momentum are set as 10^{-7} and 0.95 respectively. In each training step, our network is optimized with 300 epochs .

C. Ablation experiment

To demonstrate the effectiveness of our proposed network, we evaluate SS-Net, LS-Net and our proposed network on ShanghaiTech PartA dataset respectively. Specifically, SS-Net and LS-Net is trained on T_S and T_L separately. The evaluation results of three settings are obtained on T_A, which are shown in Table I. It is clear that our proposed network outperforms SS-Net and SL-Net in MAE and MSE. The LS-Net performs worst while SS-Net performs much better than LS-Net. These results tell that ShanghaiTech PartA has large scale variation and its count affected by the small scale heads.

D. Performance Comparison

We evaluated our network on three commonly used dense crowd counting datasets where GAK method was used to generate ground-truth density maps.

1) ShanghaiTech Dataset

This dataset proposed in [6] includes 330,165 annotations in 1,198 images which are shoot at non-uniform scenes. It is divided into two parts. Part A consists of dense crowd and Part B mainly contains sparse crowd. There are a total of 482 images with 501 average counts in Part A. Since we focus on dense crowd counting, so only Part A is considered. Several methods using Part A are taken for performance comparison. Results are shown in Table II. From it, we can see that our proposed method achieves the best performance, which is 2% lower in MAE and 13% in MSE compared with those of CSRNet [8]. For visualization purpose, several density maps estimated by our method and CSRNet are shown in Fig.3.

2) UCF_CC_50 Dataset

UCF_CC_50 [11] dataset is a collection of 50 images, which is almost the most challenging dataset due to the small amount of data with large scale variation of heads. It is noted that the count in an image varies from 94 to 4,543 and the average count is 1,280. We follow the standard settings in [11] and use 5-fold cross-validation. Several state-of-the-art methods are taken for performance comparison. Results are given in Table III, where our proposed method is superior to the comparison methods in terms of MAE and ranks second in terms of MSE.

3) UCF_QNRF

UCF-QNRF is a newly proposed dataset [17]. It contains the large amount of high-count crowd images and annotations in a wide range of scenes and viewpoints. In detail, there are 1,535 images with an average 815 count of people and the average resolution of 2013×2902. Compared with other

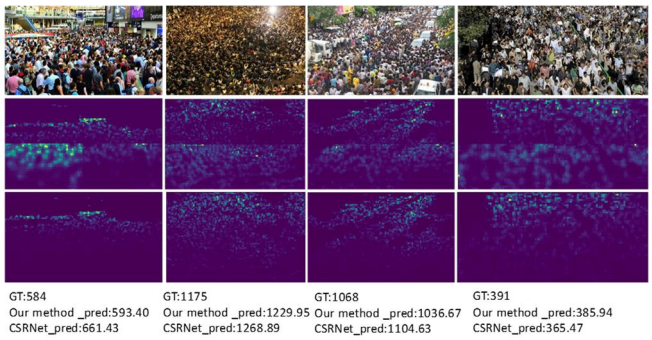

Fig. 3. Comparison on ShanghaiTech PartA. First row: test images. Second row: density maps generated by our proposed network. Third row: density maps estimated by CSRNet [8]. Clearly, results in second row are much more consistent with those in the first row. The head information in dense regions and sparse regions are all captured properly.

datasets, UCF-QNRF has the characteristic of high resolution. Experimental results are given in Table IV where our method demonstrates its superior performance compared with other methods. We have 25 counts and 18 counts gain over the latest state-of-the-art work [17] in MAE and MSE respectively.

IV. Conclusion

In this paper, aiming at alleviating the adverse effect of scale variation on counting results, we proposed a novel method for dense crowd counting where two modules with the aid of a scale-sensitive classifier are designed and trained. To the precise, a method for obtaining two scale labels of image patches is proposed. Then a sophisticatedly training approach is developed and the scale-classifier is trained to predict the scale label of the input patch and select the module for it. In the end, the density maps of input images can be estimated more accurately. The superior results on three datasets validate the effectiveness and robustness of our method.

V. Acknowledge

This paper was partially supported by Shenzhen Science & Technology Fundamental Research Programs. (No: JCYJ20160330095814461) and National Engineering Laboratory for Video Technology - Shenzhen Division, Shenzhen Municipal Development and Reform Commission (Disciplinary Development Program for Data Science and Intelligent Computing).

TABLE I. Results on ShanghaiTech PartA

Method	MAE	MSE
LS-Net	83.1	144.0
SS-Net	69.3	103.4
Proposed method	**66.8**	**99.8**

TABLE II. Results on ShanghaiTech Part A

Method	MAE	MSE
MCNN [6]	110.2	173.2
ACSCP [13]	75.7	102.7
CSRNet [8]	68.2	115.0
SANet [14]	67.0	104.5
Proposed method	**66.8**	**99.8**

TABLE III. Results on UCF_CC_50

Method	MAE	MSE
MCNN [6]	377.6	509.1
CP-CNN [15]	295.8	**320.9**
CSRNet [8]	266.1	397.5
ic-CNN [10]	260.9	360.5
Proposed method	**239.9**	367.7

TABLE IV. Results on UCF_QNRF

Method	MAE	MSE
MCNN [6]	277	426
CMTL [9]	252	514
SwitchCNN [16]	228	445
Idrees *et al*. [17]	132	191
Proposed method	**107**	**173**

References

[1] P.Viola and M.Jones. "Rapid object detection using a boosted cascade of simple features." Proceedings of the 2001 IEEE Computer Society Conference on, IEEE, pp. I-511-I-518, vol. 1. 2001.

[2] M.Rodriguez, I.Laptev, J.Sivic and J.Y.Audibert. "Density-aware person detection and tracking in crowds." IEEE International Conference on Computer Vision, pp. 2423-2430, 2011.

[3] A. C. Davies, J. H. Yin and S. A. Velastin, "Crowd monitoring using image processing," Electronics & Communication Engineering Journal, vol. 7, pp. 37-47, 1995.

[4] V.Lempitsky and A.Zisserman. "Learning to count objects in images." Advances in neural information processing systems, pp. 1324-1332, 2010.

[5] C.Zhang, H.Li, X.Wang and X.Yang, "Cross-scene crowd counting via deep convolutional neural networks." IEEE Conference on Computer Vision and Pattern Recognition, pp. 833-841, 2015.

[6] Y.Zhang, et al. "Single-image crowd counting via multi-column convolutional neural network." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 589-597, 2016.

[7] D.Onoro-Rubio, and RJ. López-Sastre. "Towards perspective-free object counting with deep learning." The European Conference on Computer Vision. Springer, Cham, pp. 615-629, 2016.

[8] Y.Li, X.Zhang, and D.Chen. "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1091-1100, 2018.

[9] Sindagi, V.A., Patel, V.M. "Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting." In 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, pp. 1–6, 2017

[10] R.Viresh, H.Le and M.Hoai. "Iterative crowd counting." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[11] H.Idrees, I.Saleemi, C.Seibert, and M.Shah, "Multi-source multi-scale counting in extremely dense crowd images." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2547-2554, 2013.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in International Conference on Learning Representations, May 2015

[13] Z.Shen, et al. "Crowd Counting via Adversarial Cross-Scale Consistency Pursuit." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5245-5254, 2018..

[14] X.Cao, Z.Wang, Y.Zhao and F.Su, "Scale aggregation network for accurate and efficient crowd counting." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[15] V.A.Sindagi and V.M.Patel. "Generating high-quality crowd density maps using contextual pyramid cnns." 2017 IEEE International Conference on Computer Vision, IEEE, pp. 1879-1888, 2017.

[16] D.B.Sam, S.Surya and R.V.Babu. "Switching convolutional neural network for crowd counting." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Vol. 1. No. 3. p. 6, 2017.

[17] H.Idrees, et al. "Composition Loss for Counting, Density Map Estimation and Localization in Dense Crowds." The European Conference on Computer Vision (ECCV), 2018, pp. 532-546.

[18] L.Boominathan, S.SS.Kruthiventi and R.V. Babu. "Crowdnet: A deep convolutional network for dense crowd counting." Proceedings of the 2016 ACM on Multimedia Conference. ACM, pp. 640-644, 2016.