Density-aware Curriculum Learning for Crowd Counting

Qi Wang, *Senior Member, IEEE*, Wei Lin, Junyu Gao, and Xuelong Li*, *Fellow, IEEE*

***Abstract*—Recently, crowd counting draws much attention on account of its significant meaning in congestion control, public safety, and ecological surveys. Although the performance is improved dramatically due to the development of deep learning, the scales of these networks also become larger and more complex. Moreover, a large model also entails more time to train for better performance. To tackle these problems, this paper firstly constructs a lightweight model, which is composed of an image feature encoder and a simple but effective decoder named Pixel Shuffle Decoder (PSD). PSD ends with a pixel shuffle operator, which can display more density information without increasing the number of convolutional layers. Secondly, a Density-aware Curriculum Learning (DCL) training strategy is designed to fully tap the potential of crowd counting models. DCL gives each predicted pixel a weight to determine its predicting difficulty and provides guidance on obtaining better generalization. Experimental results exhibit that PSD can achieve outstanding performance on most mainstream datasets while trained under the DCL training framework. Besides, we also conduct some experiments about adopting DCL on existing typical crowd counters, and results show that they all obtain new better performance than before, which further validates the effectiveness of our method.**

***Index Terms*—crowd counting, curriculum learning, neural network**

I. Introduction

WITH the development of urbanization, congested crowd scenes continually appear in squares, streets, cultural attractions, *etc.* Accompanying problems of public safety also become a new important subject. In this field, one of the typical research orientations is crowd counting, which devotes to monitoring the number of people in particular scenarios, since it may be out of control easily as it exceeds a certain threshold, and evacuation would also be a severe problem when an unusual event occurs. This kind of consciousness also dramatically promotes the development of crowd event detection [1], crowd behavior analysis [2]–[5]. Beyond the applications in public safety, the technology exploited in crowd counting also accelerates the development of other fields like space planning [6], [7], traffic monitoring [8], [9], scene understanding [10], [11], and ecological surveys [12].

To estimate the number of people in a scenario accurately, typical traditional computer vision algorithms [13]–[18] try to detect each object in images, but it does not perform well in extremely congesting scenes. Other ideas [19], [20] directly regress the count according to the extracted hand-crafted features, but these methods only work in some specific simple scenarios. Nowadays, a new fashionable scheme is estimating the density map of the corresponding crowd image, and the number is obtained by calculating integral over it. Through combining this scheme with convolutional neural networks (CNNs), considerable progress has been made in this field. Most proposed CNN-based models follow an encoder-decoder structure, as shown in Fig. 1. Firstly, crowd images are inputted into image feature encoder to produce a set of feature maps, and then these feature maps are inputted into the decoder to regress the final density map. A typical encoder is MCNN [21], which extracts crowd features through three neural branches with different receptive fields, trying to capture various density distributions. Aside from MCNN, VGG [22], ResNet [23] and DenseNet [24] which play roles in image classification are also employed in this field as encoder [25]–[27]. As for decoder, MCNN [21] uses a simple 1×1 convolutional kernel to decode density map, CSRNet [25] adopts several dilated convolutional layers as decoder, and a special decoding structure is employed in SFCN [26] for capturing perspective information.

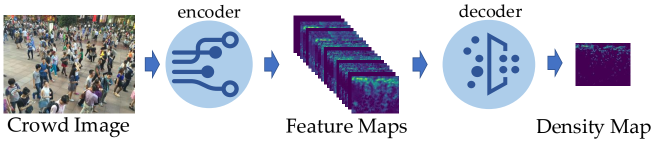


Fig. 1. **Common Crowd Counting Framework.** The crowd images is first inputted into a CNN-based encoder, generating feature maps. And the final density maps is obtained by decoding the feature maps.

Although these models and their variations achieve varying degrees of success, there is still much big promotion space in crowd counting. Firstly, the scales of models become larger and more complex, to ensure sufficient semantic information is extracted from the image and to display crowd density knowledge as much as possible. Secondly, a model with a large number of parameters entails plenty of time to be trained, and it also results in overfitting easily and performs low generalization. Aiming at these problems hereinbefore, this paper focuses on how to design a lightweight model, accelerate the training process and improve the generalization performance of models.

To establish a lightweight model, a simple but effective crowd-density decoder named Pixel Shuffle Decoder (PSD) is proposed here. PSD adopts a learning-based super-pixel

Xuelong Li is the corresponding author. This work was supported by the National Natural Science Foundation of China under Grant U1864204, 61773316, U1801262, and 61871470.

Qi Wang, Wei Lin, Junyu Gao and Xuelong Li are with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, Shaanxi, P. R. China (e-mail:crabwq@gmail.com; elonlin24@gmail.com; gjy3035@gmail.com; xuelong_li@nwpu.edu.cn). Xuelong Li is the corresponding author.

technology to decode feature maps and generates a density map with the same size as its corresponding crowd image. Its effectiveness is based on the following factor: standard encoders usually adopt pooling layers to make itself more robust, but these operations lead the predicted density map to be smaller than the original input in scale. This indicates that the produced map carries insufficient density knowledge. However, PSD is able to provide a map with more density information and does not add new convolutional layers.

To fill the gap of poor generalization and reduce training time, this paper designs a density-aware curriculum learning (DCL) strategy for crowd counting inspired by curriculum learning (CL). In previous research, CL is often adopted at the sample level, in which each training sample is given an index number as its learning difficulty for a machine learning model, and then the model is trained from simple to difficult. Bengio *et al.* [28] have proved that this strategy has the ability to guide the parameters in the learner to a better region in parameter space, which makes it gain further generalization. However, traditional curriculum learning at the sample level is not appropriate for crowd counting. On the one hand, the standard of the curriculum in this task is hard to design. Suppose we define the number of pedestrians in one image as the difficulty, splitting a large crowd image into two small images would reduce its difficulty by half, even if they have similar density distributions. On the other hand, the primary issue of crowd counting is the uneven density distribution of the crowd inside an image. It means we should concentrate on the intra-image density distribution, but not inter-image density distribution. Considering these factors, DCL is not a sample-level but a pixel-level curriculum learning method. Trivially, it uses local density as the standard of curriculum to produce local attention for an input crowd image, and gradually dampens this restriction in the training process until vanishing. It is able to accelerate convergence and enhance the accuracy of crowd counters without changing their structures. Experimental results exhibit that DCL indeed works and promotes the performance of crowd counters.

In a nutshell, the contributions of this paper lie in the following aspects:

1) A lightweight decoder PSD for crowd counting is proposed. The scale of it is small, but it can produce more elaborate density maps.
2) A pixel-level curriculum learning for crowd counting named DCL is introduced. It can improve the generalization of crowd counters without changing their network structures. Besides, this is the first work of pixel-level curriculum learning to our best knowledge.
3) Experimental results show that the proposed PSD and DCL achieves outstanding results compared with other mainstream algorithms in crowd counting. Moreover, while DCL is adopted to another crowd counters, they also perform better than before. These experiments demonstrate that our methods indeed work positively in this field.

The rest of this paper is organized as follows. Section II reviews some related works on crowd counting and curriculum learning. Section III illustrates the details of PSD and DCL. Several experiments are conducted, and their results is displayed and discussed in Section IV. Finally in Section V, our work is summarized.

II. Related Works

In this section, we review some related works about crowd counting and curriculum learning.

A. Crowd Counting

With the development of neural networks, especially convolutional neural networks(CNNs), CNN-based models [21], [25], [29]–[44] continue to refresh the record and accuracy in crowd counting.

Crowdnet [31], MCNN [21], and AMDCN [39] attempt to apply multi-column networks for scale variety. Boominathan *et al.* [31] combines two full convolutional networks (FCNs) with different numbers of layers. MCNN [21] consists of three FCNs with different kernel sizes, which leads to different receptive fields. Furthermore, the model in [39] has four FCNs, but they replace the combination of convolutional and pooling layers with dilated convolution layers, which makes it possible that the inputs and corresponding outputs have the same size. Switch-CNN [32] and the network proposed by Sindagi *et al.* [36] incorporate density level classification and density map estimation, which could be seen as a multi-task framework. Addressing scale variety, CP-CNN [33] and SAAN [34] use two networks to extract global and local context to improve the efficiency of the basal multi-column network. The model introduced by Liu *et al.* [30], Deepak *et al.* [38] and ADCrowdNet [41] introduce attention scheme into deep crowd counters. The former two input feature maps extracted from the middle layer of density map estimator to another new network to generate attention features; but Liu *et al.* [41] separate attention map generalization from density map estimation, in which way the model size is larger than the former two, but it achieves better performance. Li *et al.* [25] discover that a deeper network works better than multi-column fashion networks, so they put a series of dilated convolution layers on the top of a typical deeper network, which achieves promising results in existing crowd counting datasets. Liu *et al.* [45] combine head detection and density map estimation, and a quality network is designed to reconcile these features. MSCNN [35], SCNet [29], and SANet [37] use multi-scale network blocks to solve the scale problem. Yang *et al.* [40] and Zhou *et al.* [46] apply a multi-scale generative adversarial network to estimate high-quality crowd density maps, which infers counts more accurately.

There is no doubt that these excellent works brought a vast development space for crowd counting. However, either these networks are too small to extract useful crowd features, or are too large to be trained adequately. So in this paper, we develop a powerful decoder module that takes both accuracy and lightweight into account. The details can be found in Section. III-A.

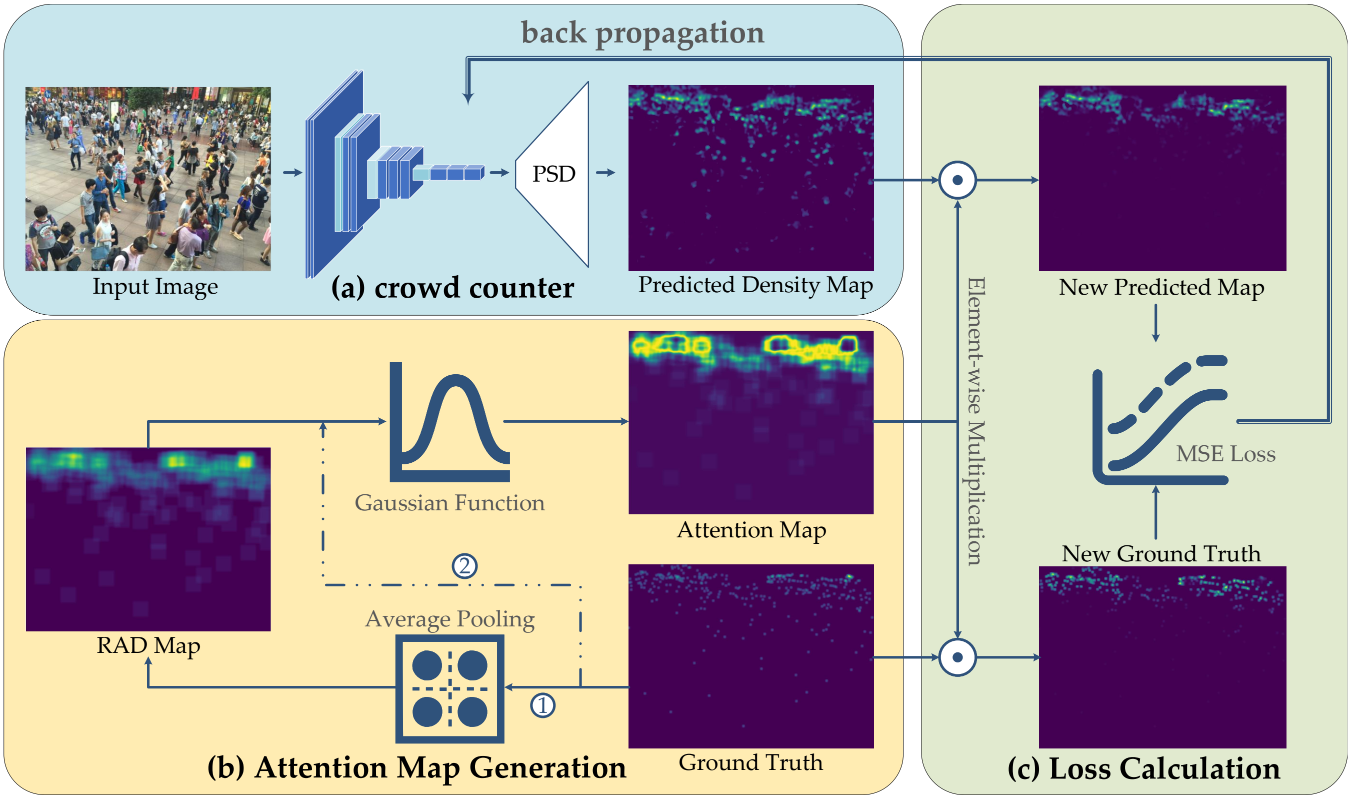


Fig. 2. **DCL algorithm flow diagram.** (a) is the process flow of any density map estimation model; (b) is the algorithm of how to define a density-aware curriculum for crowd counting task; (c) describes how to embed curriculum information into standard density map estimation neural networks, and change the training scheme of models.

B. Curriculum Learning

Curriculum learning(CL) is a learning strategy formalized by Bengio *et al.* [28]. As we all know, humans could learn better while the learning objects are not presented randomly but sorted by a meaningful order. Imitating this pattern, CL firstly give every training sample an index on behalf of its difficulty, and then train models from simple samples to hard samples.

[47]–[49] apply CL strategy on weakly-supervised object detection; Lotfian *et al.* [50] use it to maximize the efficiency of DNN in emotion recognition; Wang *et al.* [51] propose a dynamic curriculum framework, which is an adaptively sampling strategy addressing imbalanced training data. Vuddagiri *et al.* [52] introduce CL into language identification to effectively promote the generation of models and reduce environmental noise. Sarafianos *et al.* [53] apply CL to a visual attribute classification framework, which significantly boosts the performance. Saputra *et al.* [54] present a novel geometry-aware objective function as the curriculum, to train a cascade optical network for estimating monocular visual odometry. Surendranath and Jayagopi [55] create a multilevel dataset with decreasing complexity, resulting in reducing test loss significantly compared with the non-curriculum training strategy. Dong *et al.* [56] formulate a novel multi-task curriculum transfer deep learning method and achieve a notable advantage in recognizing detailed clothing characteristics. Gui *et al.* [57] adopt CL to improve the generalization of models in machine understanding and facial expression recognition.

The successes mentioned above prove that curriculum learning training strategy indeed has a positive impact on the generation and performance of machine learning models. Nevertheless, all of them are applied on the sample level, which assigns weight to training samples. For pixel-to-pixel tasks, there should be a more delicate CL method that can assign a weight to each pixel. In this paper, a pixel-level CL algorithm is designed for crowd density estimation. Several experiments are conducted to verify the advancement of our method.

III. Our Approach

This section is going to describe the proposed lightweight network and density-aware curriculum learning (DCL) in detail. As shown in Figure 2, the entire framework can be divided into three parts.

Part (a) deploys the proposed neural network, named Pixel Shuffle Crowd Counter (PSCC), in which the proposed decoder Pixel Shuffle Decoder (PSD) is employed to decode density map. Actually, in the framework of DCL, the crowd counter could be replaced by any crowd counting models, as long as it generates a density map corresponding with the input crowd image.

Part (b) is the central part of DCL framework. In short, its work is producing an attention map based on the ground truth. It provides two optional ways, whose details are introduced in III-B.

Part (c) employs the density map predicted by part (a) and the attention map generated by part (b) to produce final loss value, and use backpropagation to update the parameters in the trained crowd counting model in part (a).

At the end of this section, the algorithm is summarized, and the whole DCL training strategy is teased out.

A. Pixel Shuffle Crowd Counter

The structure of PSCC takes an encoder-decoder pattern, as displayed in Fig. 1. For the image feature encoder, it could be any neural network as long as it can extract ample semantic features of the crowd from input images. MCNN [21] is the most frequently used one in this field, and recently VGG [22] and ResNet [23] are also popular and effective due to their excellent performance in other computer vision tasks, and Li *et al.* [25] demonstrate that a deeper network works better than MCNN. PSCC adopts the first ten layers of VGG-16 as its feature encoder, as shown in Fig. 3. We do not adopt MCNN or ResNet as its backbone, since MCNN cannot extract sufficient image features, and ResNet consumes more computational time and memory.

After encoding, the extracted feature maps are inputted to the decoder, whose role is to fuse them and produce the final density map. Because the number of feature maps reaches hundreds, but it only needs one piece to denote the final density map. Previous algorithms establish varied decoding structures, but these decoders themselves also become large and complex. To construct a lightweight decoder, this paper does not focus on designing a deeper network, but exploring a structure for displaying density information as much as possible. Specifically, this paper proposes a lightweight decoder named Pixel Shuffle Decoder (PSD), whose structure is shown in Fig. 4. PSD firstly uses a Feature Compressing Module (FCM) to extract a series of density maps, and then rearrange them into a larger one through pixel shuffle operation [58], a method for super-resolving low-resolution objects into high-resolution space. Trivially, pixel shuffle is a periodic shuffling operation. Assume it return a map M_o with one channel, height of $r \times h$, and width of $r \times w$, in which case the scale of its input tensor M_i must be $h \times w$ with r^2 channels. The pixel value at $(1, x, y)$ in M_o is calculated by:

$$M_o^{(1,x,y)} = M_i^{(x\%r\cdot r+y\%r,\lfloor x/r\rfloor,\lfloor y/r\rfloor)}. \tag{1}$$

The effectiveness of PSD is based on the following: when training a crowd counter, the ground truth usually has the same scale with the corresponding image, but the scale of feature maps generated by the encoder is smaller than it due to pooling operations. Consequently, models predict a smaller density map. In past methods, the preliminary result is enlarged to fit the ground truth, and a plain up-sampling method is the Nearest-neighbor interpolation. However, this crude approach is not beneficial for a pixel-wise regression task, since it does not express sufficient density information. However, PSD is able to retain much more density knowledge and embed them into the final predicted map without increasing the number of convolutional layers in the decoder. For PSCC, there are three max-pooling layers in the employed vgg16-based encoder, which means the scale of feature maps is 1/8 of the original crowd image. So we make FCM return a feature map with 64 (8 × 8) channels for pixel shuffle operation to predict density map on the same scale as the original crowd image.

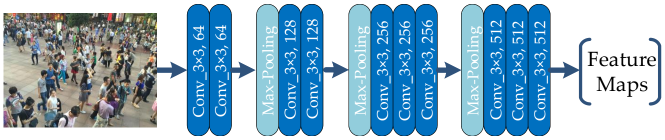


Fig. 3. **vgg-16 based encoder.** PSCC adopts vgg-16 as its image feature encoder, it only contains its first 10 layers, and discards these layers after the last max-pooling.

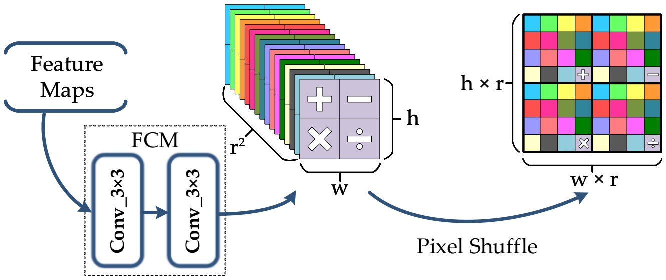


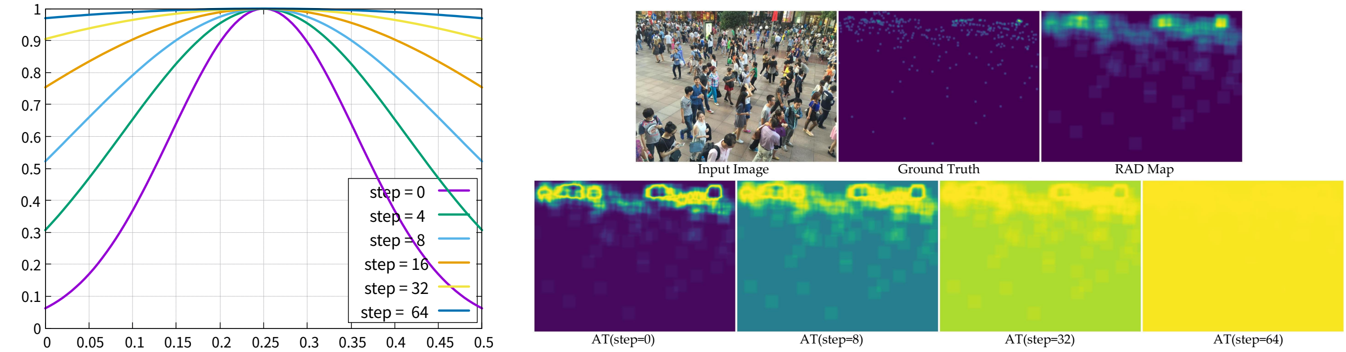
Fig. 4. **Pixel Shuffle Decoder.** The crowd images is firstly inputted into a CNN-based encoder, generating feature maps. And the final density maps is obtained by decoding the feature maps.

Actually, DCL strategy can be applied to any existing density map estimation models addressing crowd counting, as long as these models take a crowd image as input and output its corresponding density map. This paper also presents some experiments through other typical models like MCNN, CSRNet, and SFCN. Details could be found in IV-G.

B. Attention Map Generation

Density-aware curriculum learning (DCL) strategy has the ability to prompt the performance of crowd counters without changing their network structures. As we have described in Section I, traditional sample level curriculum learning (CL) do not satisfy crowd counting since crowd counting is a pixel-level regression model. However, DCL is a pixel-level CL strategy. Because it assigns each pair of corresponding pixels in predicted map and ground truth a weight through an attention map when calculating the loss value, which could be seen as the curriculum difficulty. It is also important to note that this attention map only works during the training phase, like traditional curriculum learning, and it would not work while given novel crowd images.

As shown in Fig. 2(b), in order to generate the density map, DCL equips two operations and leverages ground truth as input. These two operations are average pooling and Gaussian function respectively. To be specific, it firstly generates a region average density (RAD) map according to ground truth through the average pooling operation, and then produces an attention map through the Gaussian function, which gives more weight to simple pixels. The impact of DCL becomes slack gradually during training process. In the following part, how DCL works is introduced in detail.



(a) (b)

Fig. 5. (a) illustrates the attention variation with different RAD and step. (b) demonstrates an image, its corresponding ground truth, RAD map, and difficulty maps in different phases. Each value in ground truth is enlarged by 100, and set $r = 32$, $\mu = 0.5, \alpha = 0.02, \beta = 0.15$, and $\gamma = 1$

In general, a crowd image I has N labeled heads $\boldsymbol{x} = \{x_1, \cdots, x_N\}$, in which each x_i is a 2D coordinate and represents an object location. The density map corresponding to I is obtained by convolving a Gaussian kernel G_σ with $\boldsymbol{x}$:

$$D = G_\sigma(\boldsymbol{x}), \tag{2}$$

where σ represents the spread parameter. DCL defines the curriculum according to RAD, which can be formulated by

$$RAD_r(x, y, D) = \frac{1}{(2r+1)^2} \sum_{x-r}^{x+r} \sum_{x-r}^{x+r} D_{i,j}, \tag{3}$$

in which r is a super parameter representing the local region size, and D denotes the density map obtained through Eq (2). It is evident that $RAD_r(x, y, D)$ means the average density value of a rectangular region which takes (x, y) as the center, with a width of $2r + 1$ pixels. Eq (3) could be easily applied through average pooling operation, with kernel size of $2r+1$, stride of 1 and padding value of r. In the remainder, we use $R_r(D)$ to represent the RAD map obtained through adopting Eq (3) to all pixels in D.

After figuring out $R_r(D)$, the attention map (AT) is defined through Gaussian function, which is formulated as:

$$AT_{\mu,\delta,\gamma}(R) = \gamma \cdot \exp\left[-\frac{(R-\mu)^2}{\delta^2}\right], \tag{4}$$

in which R represents $R_r(D)$ obtained through Equation (3), μ is the center position of the Gaussian curve, δ is another parameter controlling its width, and γ represents the height of the curve's peak. Obviously, $AT_{\mu,\delta,\gamma}(R)$ gives more attention to those pixels whose RAD is close to μ. This implies that DCL sets those pixels whose value is close to μ as the simplest sample points in each single step of curriculum learning.

Besides the impact of μ, δ is employed as a controller, whose job is controlling the curriculum changes during the training phase. In this paper, it grows linearly with training steps:

$$\delta = \alpha \cdot \text{step} + \beta, \tag{5}$$

in which α denotes growth rate, and β is its initial value.

Through Equation($3 \sim 5$), the final attention map is formulated as:

$$AT_{r,\mu,\alpha,\beta,\gamma}(D, \text{step}) = \gamma \cdot \exp\left[-\left(\frac{R_r(D)-\mu}{\alpha \cdot \text{step} + \beta}\right)^2\right] \tag{6}$$

To better understand the concept of Eq. (6), Fig. 5 presents the visualization of attention value changing curves and an example of how the attention map changes during training. In Fig. 5, it assumes $\mu = \frac{1}{2}R_r(D)$, which makes the crowd counter focus more on these crowd spots which have enough distinctive features, and pay less attention to extremely sparse and dense part. In the beginning, most pixels in AT except a few is close to 0, after certain training steps, these values gradually increase, until all of them are close to 1(γ). This can be considered as a process of increasing the curriculum complexity in curriculum learning.

Some experimental results show that the model may perform better without average pooling operation, so Fig. 2 (b) provides alternative accesses, with or without pooling operation. Actually, it could be seen as pooling size of 1×1 in the latter case, so this paper does not highlight the difference.

C. Loss Function

The way that DCL influences the parameters of the trained crowd counting model is to change its loss function in different training phases. Trivially, while adopting DCL training framework, the loss function is formulated as follows:

$$\mathcal{L}(\Theta)_{\text{step}} = \frac{1}{2N} \sum_{i=1}^{N} \|AT(D_i, \text{step}) \odot (X_i - D_i)\|_2^2, \tag{7}$$

where $\odot$ represents element-wise multiplication, Θ is the parameters of a crowd counting neural network, N is the number of samples in the training dataset, X_i and D_i denote the i-th input image and the corresponding ground truth respectively, and step represents training period. From Eq (6), the following equation can be obtained:

$$\lim_{\text{step}\to\infty} AT(D_i, \text{step}) = \gamma, \tag{8}$$

which means all values in AT are going to approach γ, as shown in Fig. 5(a). And the following result is derived:

$$\begin{aligned}\lim_{\text{step}\to\infty}\mathcal{L}(\Theta)_{\text{step}} &= \frac{1}{2N}\sum_{i=1}^{N}\|\gamma\cdot(X_i-D_i)\|_2^2 \\ &= \frac{\gamma}{2N}\sum_{i=1}^{N}\|X_i-D_i\|_2^2\end{aligned}\quad . \tag{9}$$

Eq (9) suggests that the DCL will degrade the loss function to a normal one (Eq (7)) since γ is a constant.

D. Algorithm Flow

According to the above analysis, the algorithm of DCL could be outlined in Algorithm 1:

Algorithm 1 Density-aware Curriculum Learning

```
Input: Training dataset $D = \{(I_1, D_1), \cdots, (I_N, D_N)\}$,
    density map estimator $\mathcal{F}(\Theta)$,
    super parameters of DCL $r, \mu, \alpha$, $\beta$ and $\gamma$
Output: $\arg\min_{\Theta}\sum_{i=1}^{N}(\mathcal{F}(\Theta; I_i) - D_i)^2$
1: define average pooling operator $\mathcal{AP}_r$
2: define Gaussian function $G_{\mu,\gamma}$
3: initialize the parameter of density map estimator $\Theta$
4: initialize training step: $s \leftarrow 0$
5: while not converged do
6:     update step: $s \leftarrow s + 1$
7:     for $k \leftarrow 1$ to $N$ do
8:         $R_k = \mathcal{AP}_r(D_k)$
9:         $AT_k^s = G_{\mu,\gamma}(R_k; \alpha s + \beta)$
10:        predicted result: $P_k = \mathcal{F}(\Theta; I_k)$
11:        new predicted map: $P_k^s = AT_k^s \cdot P_k$
12:        new ground truth: $D_k^s = AT_k^s \cdot D_k$
13:        $\Theta^* = \arg\min_{\Theta}\|P_k^s - D_k^s\|_2^2$
14:    end for
15: end while
16: return $\Theta^*$
```

Given training dataset $D = \{(I_i, D_i)|0 \le i \le N\}$ and a crowd density estimator $\mathcal{F}(\Theta)$, the goal is obtaining suitable Θ to minimize the sum of $(\mathcal{F}(\Theta; I_i) - D_i)^2$ for all data pairs. In step 1-4, some operators are set by given super parameters: the average pooling operator defined by r (Eq (3)), the Gaussian function defined by μ and γ (Eq (4)), the initial parameters Θ of crowd counter, and the training step s. Step 5 assesses whether the model $\mathcal{F}(\Theta)$ is converged, generally it is a loop with certain steps, and is assessed manually (300 in our experiments). Step 7 means sampling from the training dataset to obtain data pair (I_k, D_k), and step 8-13 is the training process. In step 8, RAD map is obtained through Eq (3); in step 9, attention map is obtained by Eq (6); step 10 represents adopting $\mathcal{F}(\Theta)$ to predict density map; in step 11-12, predicted density map and ground truth is multiplied to attention map. Finally in step 13, Euclidean distance between weighted predicted map and weighted ground truth is calculated, and Θ is updated through backpropagation.

IV. Experiments

In this section, the experiments we have conducted about PSCC and DCL are introduced in detail. Firstly, the experimental settings and evaluation are described. Secondly, we perform an ablation study on ShanghaiTech Part A dataset [21] to analyze the effect of different parts of DCL and PSCC. Thirdly, the experimental results of how the proposed methods perform on some mainstream datasets are reported. Finally, we apply DCL on other typical crowd counting models (MCNN [21], CSRNet [25], and SFCN† [26]), which proves that the proposed DCL training strategy can boost all crowd counting models.

A. Experiment Evaluation

By following existing works, we evaluate our method with both the absolute error (MAE) and the mean squared error (MSE), which are formulated as follows:

$$\begin{aligned}\text{MAE} &= \frac{1}{N}\sum_{i}^{N}|\hat{C}_i - C_i|, \\ \text{MSE} &= \sqrt{\frac{1}{N}\sum_{i}^{N}|\hat{C}_i - C_i|^2},\end{aligned} \tag{10}$$

in which N denotes the number of samples in the test dataset, $\hat{C}_i$ and C_i represent the predicted count value and the count label of the i-th sample respectively. Besides, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity in Image (SSIM) [59] are applied to measure the quality of density maps.

B. Implementation Details

As described in Eq (6), there are five super parameters (including r, μ, α, β, and γ) to be set before training a crowd counter. Since the crowd density and the number of images vary from one dataset to another, these parameters are also different. For example, a dataset with more pictures leads to a larger α, since it may converge through fewer epochs, but DCL could not work adequately before the δ (in Eq (5)) is large enough. Average crowd density has an impact on r and μ. The former determines the RAD map. It is obvious that a larger r could dilute the RAD map, so it is not appropriate for images with sparse crowd. The latter determines which RAD value should be paid more attention on. In our experiments, μ is calculated by:

$$\mu = \frac{1}{2N}\sum_{i=1}^{N}\max(R_r(D_i)), \tag{11}$$

which means μ is half of the average peak value in each ground truth map, and N is the number of samples in the dataset. μ defines which pixel should own most attention, Eq (11) signifies both the most sparse or densest region are not appropriate choices. This is according to two factors: firstly, RAD in sparse regions is too small, which would lead lots of neurons to be inactivated; secondly, the dense regions suffer from a shortage of image features, which would make DCL bypassed.

TABLE I
DIFFERENT SETTINGS OF DCL IN DIFFERENT DATASET.

dataset	r	μ	α	β	γ
WorldExpo10 [61]	0	0.683	0.2	0.3	2
SHT A [21]		1.482	0.05		
SHT B [21]		1.02	0.05		
UCF-QNRF [27]	1	2.55	0.2		
GCC [26]		2.1	0.3		

TABLE II
RESULTS OF DIFFERENT PART ON SHANGHAITECH PART A

Methods	MAE	MSE
VGG + REL + NNI	72.12	115.26
VGG + FCM + NNI	68.58	112.34
VGG + FCM + PixelShuffle (PSCC)	66.82	109.35
PSCC + DCL	**64.97**	**107.96**

Finally, according to the different characteristics of different datasets, this paper list the detailed setting in TABLE I for future reference. For simplicity, all experiments are conducted under C^3 Framework [60], an open-source PyTorch code for crowd counting.

C. Ablation Experiments on ShanghaiTech Part A

In this section, we exhibit some ablation experiment results for a better understanding of our proposed methods. These experiments are conducted on the ShanghaiTech Part A dataset, which contains 300 images for training and 182 images for testing. Most images in it are collected from the Internet, so they vary in image size, crowd density, and even photorealistic style. As illustrated in TABLE II, four step-wise models are constructed to illustrate each part's performance improvement. All these models leverage VGG-16 as an image feature encoder. The first one is a baseline, which takes a Regression Layer (REL, a convolutional layer with 1 × 1 filters) and Nearest-Neighbor Interpolation (NNI) operator as the decoder. Based on it, the second model replaces REL with FCM, and the performance takes a big step. The third one is our PSCC, which takes our proposed PSD as the decoder. The last one applies DCL strategy to train PSCC, which achieves the best of MAE (64.47) and MSE (107.96).

Fig. 6 demonstrates the visualization of some test crowd images. The first and the second row display the original crowd image, and corresponding density map labeled manually. The rest four rows post the density maps predicted by the above four step-wise models. From Fig. 6, we can see that our PSCC produces better density maps compared with two baseline models. At least its predicted maps do not contain plenty of pixel blocks like the previous two rows. Although PSCC achieves comparative results, more details are generated after applying DCL on it, as shown in the red box. In brief, FCM decodes more precise density maps than baseline, pixel shuffle operation trims the blocks generated by the upsampling module, and DCL elevates the overall performance of PSCC from the training strategy aspect.

1) Effect of FCM: The second row in TABLE II reports the results of VGG + FCM. Obviously, it produces smaller estimation errors (MAE: 67.51, MSE: 109.06) than the previous baseline (MAE: 69.96, MSE: 113.61). On one side, the decoder containing FCM owns more learnable parameters, which leads the decoder to be more precise. On the other side, it also has a deeper network structure and larger receptive fields. Instead, the baseline only uses a regression layer with 1×1 convolutional structure, which is too straightforward to deconstruct and analyze these crowd image feature maps.

2) Effect of Pixel Shuffle: While replacing nearest-neighbour interpolation with pixel shuffle operation, PSCC achieves smaller mean absolute error (66.82). Pixel shuffle provides two significant benefits. Firstly, it helps the crowd counter produce a more precious density map. While upsampling by NNI, each pixel is amplified as a square spot with the same value. However, PSD predicts a particular density value for each pixel. So PSD is able to produce a map with more density information. Secondly, these feature maps which are assigned weights close to zero by NRM may not be considered for the final density map, leading to a waste of resources. However, pixel shuffle can leverage each feature map generated by FCM, and fully activate each filter before pixel shuffle operation.

3) Effect of DCL: DCL is the main point that this paper wishes to emphasize. It could guide the crowd counter toward a better region in parameter space where the model has more generalization ability during training. This potential advantage pushes its limit and further diminishes the estimation errors (MAE: 64.47, MSE: 107.96). Comparing with the model trained without DCL, MAE, and MSE are reduced by 2.35 and 1.39. By the way, it is evident from Fig. 6 that DCL can display better details, and these details prompt the predicted density map closer to the ground truth.

TABLE III
MAE AND MSE ON SHANGHAITECH DATASET.

Method	Part A		Part B	
	MAE	MSE	MAE	MSE
Zhang *et al.* [62]	181.8	277.7	32	49.8
MCNN [21]	110.2	173.2	26.4	41.3
Switch-CNN [32]	90.4	135	21.6	33.4
CSRNet [25]	68.2	115	10.6	16
PCC Net [63]	73.5	124	11	19
SCAR [64]	66.3	114.1	9.5	15.2
PACNN [65]	66.3	106.4	8.9	13.5
ASD [66]	65.6	**98.0**	8.5	13.7
SFCN [26]	–	–	9.4	14.4
TEDNet [67]	**64.2**	109.1	8.2	**12.8**
PSCC+DCL	65.0	108.0	**8.1**	13.3

D. Results on Mainstream Dataset

In this part, this paper compares PSCC with several existing robust algorithms addressing crowd counting, including MCNN [21], Switch-CNN [32], CSRNet [25], SFCN [26] and so forth. These methods are evaluated on four entirely different datasets, ShanghaiTech dataset A/B [21], UCF-QNRF [27], GCC [26], and WorldExpo'10 [62].

1) Results on ShanghaiTech: ShanghaiTech dataset [21] is composed of two parts, A and B. Part A has been introduced in

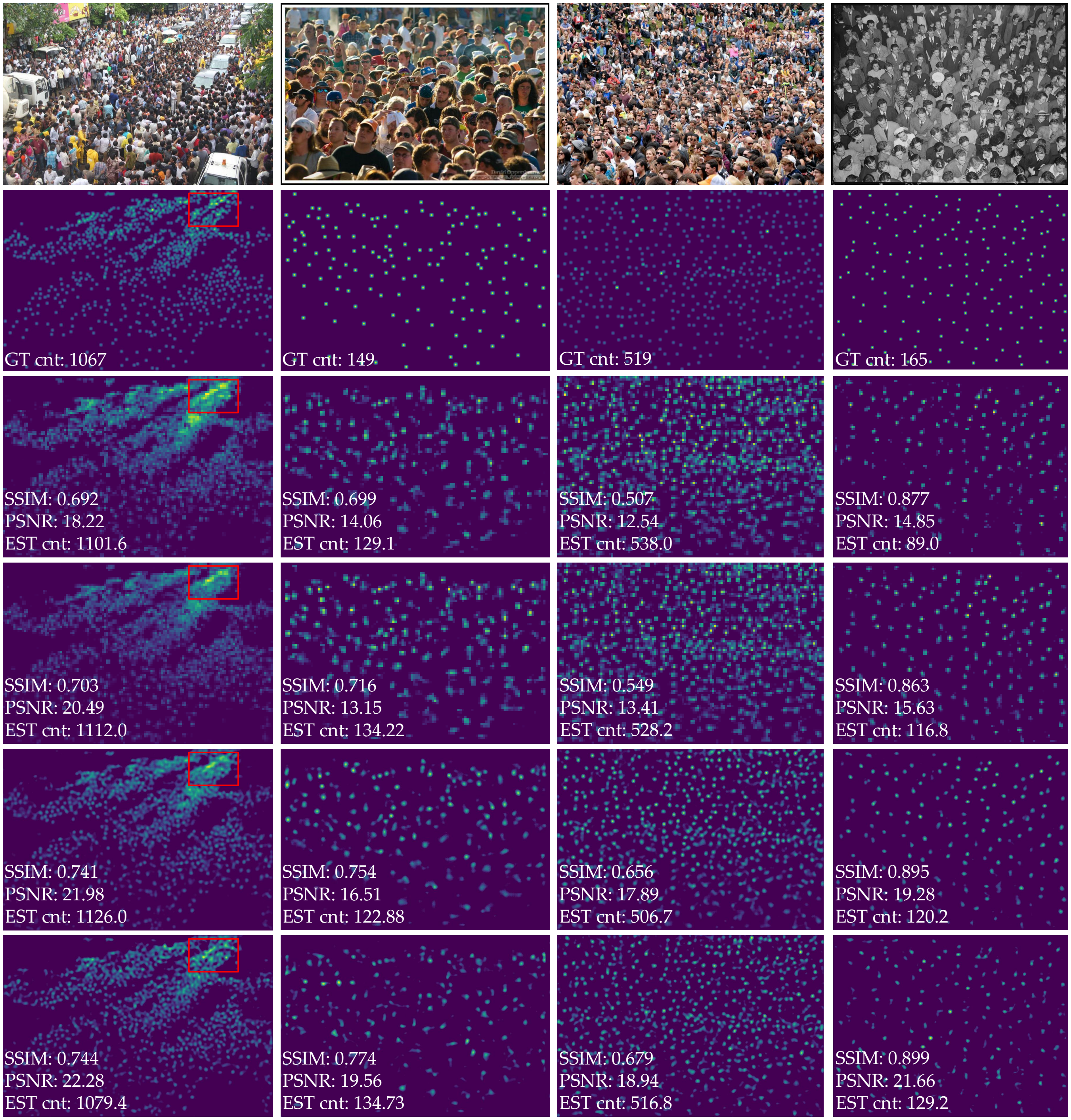


Fig. 6. **Typical examples of step-wise models on Shanghai Tech Part A dataset.** The first row shows the inputted crowd images; the second row displays the ground-truth density maps; The other four rows demonstrate the results of VGG+REL+NNI, VGG+FCM+NNI, PSCC and PSCC + DCL. *GT cnt* means the ground truth count and *EST cnt* denotes the estimated count.

TABLE IV
MAE AND MSE ON UCF-QNRF DATASET.

Methods	MAE	MSE
Idrees *et al.* [68]	315	508
MCNN [21]	277	426
Switch-CNN [32]	228	445
CSRNet [25]	122	202
PCC Net [63]	148	247
CAN [69]	**107**	183
SCAR [64]	122	207
ASD [66]	123	200
SFCN [26]	124	203
TEDNet [67]	113	188
PSCC+DCL	108	**182**

TABLE V
MAE AND MSE ON WORLDEXPO'10 DATASET

Methods	S1	S2	S3	S4	S5	Mean
Zhang *et al.* [62]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN [21]	3.4	20.6	12.9	13	8.1	11.6
Switch-CNN [32]	4.4	15.7	10.0	11.0	5.9	9.4
CSRNet [25]	2.9	11.5	8.6	16.6	3.4	8.6
PCC Net [63]	1.9	18.3	10.5	13.4	3.4	9.5
CAN [69]	2.9	12	10	7.9	4.3	**7.4**
SCAR [64]	1.9	13.8	9.6	29.8	3.9	11.8
ASD [66]	2.5	14.2	**7.1**	**7.4**	3.8	7.1
SFCN [26]	1.8	17.5	11.1	13.5	3.0	9.4
TEDNet [67]	2.3	**10.1**	11.3	13.8	**2.6**	8
PSCC+DCL	**1.8**	16.2	9.2	25.0	2.8	11.0

Section IV-C. Unlike Part A, Part B is collected in urban areas from a drone perspective, so the images in it are homogeneous. It contains 716 images, 400 for training, and 316 for testing. All these images have the same resolution of 768 × 1024.

The MAE and MSE are reported in TABLE III. According to it, PSCC+DCL achieves competitive results compared with most methods on this dataset. In Part A, it estimates the smallest error under MAE (65.0) and the third-smallest under MSE (108.0). And in Part B, it achieves the most amazing results (MAE of 8.1 and MSE of 13.3).

2) Results on UCF-QNRF: UCF-QNRF [27] is the most congested crowd counting dataset. It contains 1200 images for training and 335 for testing, and all pictures in it are collected from the Internet. The average count is 815, which is 1.6 times larger than ShanghaiTech A, and 6.6 times larger than ShanghaiTech B. Besides, the number of images is nearly twice as large as ShanghaiTech.

As shown in TABLE IV, this paper compares PSCC+DCL with ten different methods. For MAE, our approach (MAE: 108) only performs worse than CAN (MAE: 107), but the difference is merely 1.0. As for MSE, PSCC+DCL is the best crowd counter (MSE: 182) among these compared methods.

3) Results on WorldExpo'10: WorldExpo'10 is a cross-scene dataset, and the images in it are taken from 108 different camera perspectives in Shanghai EXPO 2010. It contains 3,980 crowd images and 199,923 labeled heads. However, it is not an extremely congested crowd counting dataset, since the number of heads in a single image does not exceed 253. The training set consists of 103 scenes, and the test set is composed of another five scenes.

TABLE VI
MAE AND MSE ON GCC DATASET

Methods	random		cross-camera		cross-location	
	MAE	MSE	MAE	MSE	MAE	MSE
FCN [26]	42.3	98.7	61.5	156.6	97.5	226.8
MCNN [21]	100.9	217.6	110.0	221.5	154.8	340.7
Switch-CNN [32]	115.1	18.4	115.1	244.1	142.1	324.4
CSRNet [25]	38.2	87.6	61.1	134.9	92.2	220.1
PCC Net [63]	32.0	230.9	55.2	303.6	**85.7**	469.1
CAN [69]	38.0	81.2	57.1	123.1	89.5	236.5
SCAR [64]	31.7	**76.8**	55.8	135.3	87.2	220.7
ASD [66]	37.1	81.9	58.7	141.8	86.6	**213.5**
SFCN [26]	36.2	81.1	56.0	**129.7**	89.3	216.8
PSCC+DCL	**31.3**	83.8	**53.1**	137.9	89.0	218.4

TABLE V displays the estimation error of PSCC+DCL and some super-duper algorithms. S1-S5 represent the MAE of these 5 test scenes, and *Mean* is the average value of them. PSCC+DCL does not work better than other models, and its mean error is only smaller than MCNN, SCAR, and [62]. However, it achieves the best result in S1 (1.8), which is the same as SFCN, and it is the second-best model in S5 (2.76, TEDNet: 2.6).

4) Results on GCC dataset: GCC [26] is a synthetic dataset that is collected and labeled freely. It is also the largest available crowd dataset. The count in it is from 0 to 12865, and the mean count is 501, which is close to ShanghaiTech Part A but varies more in the scale of the crowd. In [26], Wang *et al.* design three types of experiments: random splitting, cross-camera splitting, and cross-location splitting. Details could be found in [26]. This paper does not introduce it due to the limited space, and the experiments of adopting PSCC+DCL on GCC are also conducted following the above three schemes.

The estimation errors are displayed in TABLE VI, and PSCC+DCL obtains competitive results. While GCC is split randomly, SCAR [64] estimates the smallest MSE (76.8), and there is a big gap between it and PSCC (83.8). Nevertheless, PSCC+DCL estimates the smallest MAE (31.3). In the cross-camera splitting scheme, PSCC+DCL also has the top result in MAE (53.1), though it does not achieve the best result in MSE (137.9 versus 129.7 of SFCN). As for the cross-location splitting, PSCC+DCL gets the third-best result in MSE (218.4), higher than ASD and SFCN. PCC Net performs the best on MAE, but its MSE result is always the worst. Compared with PCC Net, our model does not show a better MAE performance, but it produces more balance and credible results.

5) Discussion: According to the aforementioned experimental performance description, this part discusses the strengths and weaknesses of these methods. MCNN is the most classical one of crowd counting, even though it makes a big push to this field, its number of parameters is not sufficient, and its design of network structure is not mature. Deb *et al.* [25] conducted some experiments and concluded that the feature maps produced by three different columns are similar. Although Switch-CNN puts a classifier before sending image patches into different columns, it does not put much training parameters on the regression part. The number of parameters in the classifier is 110 times of density map

TABLE VII
MAE BASED ON DIFFERENT POOLING RADIUS r

Pooling Radius (r)	0	1.0	2.0	4.0	8.0
SHHT-B	**8.08**	8.12	8.12	8.27	8.31
UCF-QNRF	108.45	**107.74**	110.48	111.57	112.13

TABLE VIII
MAE, MSE, LOSS VALUE BASED ON DIFFERENT γ

Peak Value (γ)	0.5	1.0	2.0	4.0	8.0
MAE	14.66	9.86	**8.08**	8.13	8.29
MSE	25.89	16.07	**13.25**	13.66	13.64
Loss	0.0060	0.0050	**0.0048**	0.0049	0.0050

estimator. So Switch-CNN obtains better results but does not touch the key point. CSRNet and SFCN are representations of deeper neural networks in crowd counting. They further elevate the neural network's performance in density estimation, but their structures are too complex to overfit. ASD is an interesting model, which is inspired by the switcher in Switch-CNN, but it not only uses VGG-16 as the backbone of switcher, but also adopt it as the feature extractor of estimator. These models all estimate outstanding results. However, they need a larger training dataset and an appropriate training strategy to lead them with better converge to be promoted. By the way, there should be an upsampling algorithm devoting to interpolation for more refined density map, due to the pooling operation of feature extractor. For our method, PSCC adopts PSD to decode and regress a pixel-to-pixel correspondence crowd density map, and DCL leads the crowd counter to obtain more stable and generalized parameters.

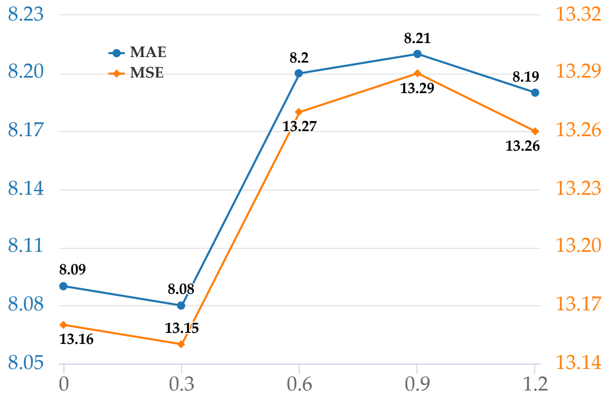


Fig. 7. the changing situation of MAE according to different β on SHHT-B. ($\beta = 10^{-5}$ to avoid Eq (6) dividing 0 when setting $\beta = 0$ in this figure)

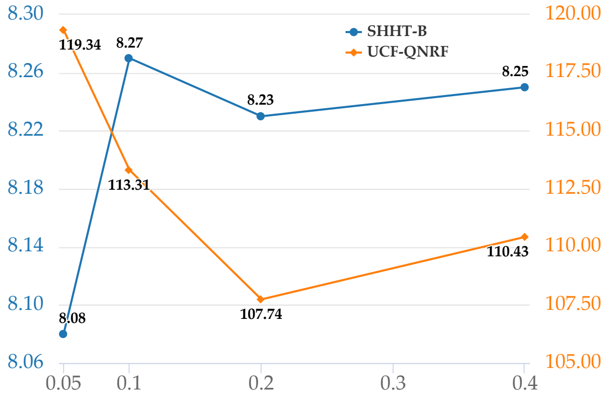


Fig. 8. the changing situation of MAE according to different α on SHHT-B and UCF-QNRF.

E. Discussion on Super Parameters

For further understanding of our algorithm, this part discusses the super parameters in Density-aware Curriculum learning.

1) Pooling Radius: Pooling radius is the r in Eq (3), representing the local region size. It also refers to which way is selected in Fig. 2(b). To further clarify which way should be used for different scenarios, here we conduct a series of experiments with different radius on SHHT-B and UCF-QNRF. TABLE VII demonstrates the relationship between radius and MAE. This table shows that PSCC performs best when $r = 0$ on SHHT-B, and best when $r = 1$ on UCF-QNRF. Why the same model needs different pooling radius to perform better? It is because of the different crowd density of these two datasets. UCF-QNRF is more congest than SHHT-B, so applying a pooling operation could dilute the local density and make the RAD map smoother. Even the radius is 1, the pooling area would be 9 according to Eq (3). The table also shows there is an upper bound for exceeding congest scenes. When the radius is over the bound, the performance of crowd counter may become worse.

2) Peak Value: Peak value is the maximum attention in DCL, which is γ in Eq (4) and Eq (6). Since peak value has no intuitive relation with crowd density, we set it as 2 for all datasets. In this part, we only conduct experiments on SHHT-B for quick verification. As shown in TABLE VIII, PSCC obtains the best results while $\gamma = 2$. More remarkable, when $\gamma \leq 1$, the performance is worse than the original random training strategy. This is because lower weight can not produce ideal loss value. If the loss value is l before adopting DCL, it would be less than l while applying a γ that is not bigger than 1, according to Eq (7). On the contrary, it is not appropriate to give a larger γ, either. A larger γ could produce larger loss value, which leads the model to vacillate in parameter space. Actually, this can be solved by adjusting the learning rate. However, set a suitable peak value ($1 < \gamma < 4$) is more accessible than adjusting the learning rate during the training process. Moreover, we also exhibit the loss value when PSCC reaches the best results in TABLE VIII, verifying that an appropriate γ can make the crowd counter converge better.

3) Controller δ: Referring to Eq (5), δ works as a controller to control the curriculum changes during the training phase, and it is a linear function about step with parameter α and β. β is the initial value of controller δ, and it determines the starting point of DCL. A very large β can degrade DCL to normal random training strategy. As shown in Fig. 7, DCL usually works well when β is small, and the performance is worse with the increase of β but would not be worse than standard training strategy. Another part of δ is α, which is the growth rate of DCL, controlling the gross value of an attention map. In our idea, α is related to the size of the dataset and

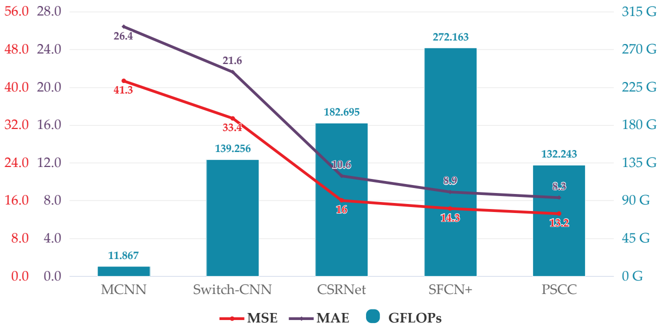


Fig. 9. Comparison of MAE, MSE, GFLOPs between MCNN, Switch-CNN, CSRNet, SFCN+ and our PSCC.

TABLE IX
THE EFFICIENCY TEST OF MAINSTREAM MODELS

Network	MCNN	Switch-CNN	CSRNet	SFCN†	PSCC
Size(M)	**0.133**	15.023	16.263	38.597	8.963
Speed(fps)	**129.0**	38.2	26.1	8.8	44.6
GFLOPs(G)	**11.867**	139.256	182.695	272.763	132.243

how many epochs the trained model can converge when given an individual attention map for each training image pair. With the above analysis, we conduct experiments about α on SHHT-B and UCF-QNRF. The former has 400 images for training, and the latter owns 1200 training images. The experimental results are shown in Fig. 8. With certain training steps, the crowd counter could see more samples on UCF-QNRF. So DCL should give a larger growth rate on this one. This is why UCF-QNRF could obtain the best result (MAE: 107.74), but get an unsatisfactory performance on SHHT-B when $\alpha = 0.2$. As for SHHT-B, the growth rate should be adjusted to a smaller one for a smaller MAE, and the experiment shows that while α is close to 0.05, the best result is obtained (MAE: 8.08).

F. Efficiency Analysis

PSCC positions itself as a lightweight crowd counter. To detailedly analyze our model, we employ GLOPs as the standard to compare PSCC with other mainstream models (MCNN, Switch-CNN, CSRNet, and SFCN+). GLOPs is the number of multiplications and additions while inputting an image to a network. In our experiment, we set the size of input as 576 × 768. Not only GFLOPs, but we also compare the test time on the same machine to verify the efficiency of PSCC and other models. As shown in Fig. 9, PSCC can achieve the best results with the second largest model size. MCNN is the smallest one, but its performance is the worst one. TABLE IX demonstrates efficiency evidence in more detail, from which we can point out that: PSCC produces more delicate and accurate results with a lightweight model, and less computation.

G. Applying DCL on Typical Mainstream Networks

For further revealing that DCL training strategy could boost the crowd counter's performance, this section introduces some experimental results of applying DCL on other mainstream methods. Specifically, these experiments replace the PSCC in Fig. 2 with other crowd counters to see whether their performance is elevated. These methods are MCNN [21], CSRNet [25] and SFCN† [26]. MCNN adopts a multi-column neural network as the encoder and NRM as the decoder; CSRNet deploys VGG-16 as the encoder, and some dilated convolutional layers as decoder; SFCN† employs ResNet-101 as the backbone and a spatial decoding structure as the decoder. These algorithms vary in model depth, design philosophy, and complexity, so whether they perform better than before is proof of DCL's effectiveness. As for dataset, this paper also selects two representative datasets, ShanghaiTech (SANet and UCF-QNRF, since the details of these two datasets, have been introduced in IV-D, this section does not duplicate here.

TABLE X
ESTIMATION ERRORS OF SOME MAINSTREAM ALGORITHMS

Methods		SHHT B		UCF-QNRF	
		MAE	MSE	MAE	MSE
MCNN [21]	baseline	26.41	41.3	277	426
	C3F	25.51	41.31	257.09	389.86
	DCL	**24.09**	**38.47**	**208.2**	**321.74**
CSRNet [25]	baseline	10.6	16	–	–
	C3F	10.82	16.76	119.05	196.3
	DCL	**8.94**	**14.81**	**115.9**	**189.06**
SFCN† [26]	baseline	8.9	14.3	114.8	192
	C3F	8.55	14.12	115.7	194.6
	DCL	**7.55**	**12.96**	**107.1**	**189.1**

TABLE X demonstrates the experimental results of conducting the above algorithms on ShanghaiTech Part B and UCF-QNRF. In the second column of TABLE X, baseline means the results presented by the original published paper, C3F [60] represents the results reproduced under C^3-Framework, and the row led by DCL exhibits the estimation errors after applying DCL on its corresponding C3F vision. All these three models estimate smaller errors on both datasets under the DCL training strategy. To be specific, MAE and MSE of MCNN decrease about 5.6% and 6.9% respectively on ShanghaiTech Part B and decline more on UCF-QNRF (MAE: 19%, MSE 17.5%). Not only MCNN, but also the results of CSRNet on ShanghaiTech Part B reduce 17.4% and 11.6%, and fall by 2.6% and 8.3% on UCF-QNRF, respectively. SFCN† is the deepest neural network among these three models. It is worth mentioning that DCL prompts SFCN† to achieve state-of-the-art performance (SHHT B: 7.55/12.96, UCF-QNRF: 107.1/189.1), which is better than PSCC+DCL proposed in this paper.

This paper also records MAE's changing curves during the training phase, which is shown in Fig. 10. More to the point, Fig. 10 presents the changing situation of MAE on the corresponding validation set while training MCNN on UCF-QNRF. Similarly, C3F and DCL represent different training schemes. The vertical axis value is not the real MAE but its logarithm, due to the high starting point of DCL. From the figure, we can conclude that even DCL estimates huge MAE in the beginning, it drops rapidly and is comparable with C3F in about 30 epochs. After that, it is evident that DCL has guided MCNN to a region where it can obtain better adaptability in

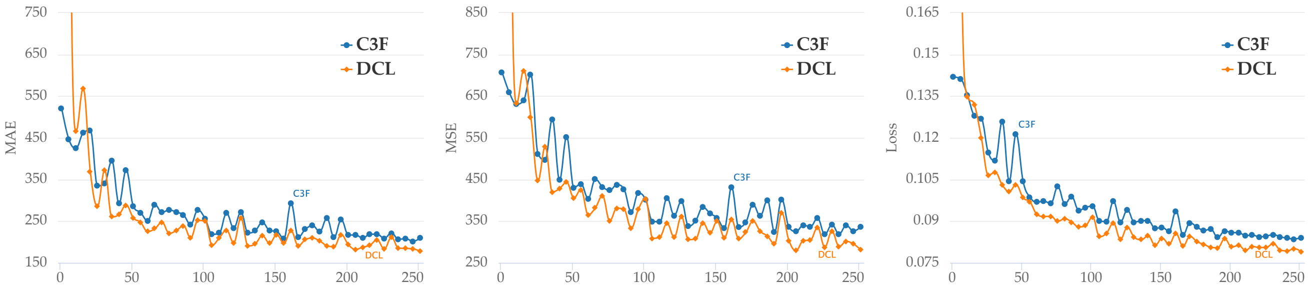


Fig. 10. **MAE, MSE and Loss Changing Curve.** These three figures demonstrate the changing situation of MCNN on QNRF during training phase.

the parameter space. So the curve on behalf of DCL is always under the curve of C3F, which also confirms the effect of DCL.

This section introduces experimental results while training another crowd counter under DCL strategy. Better performances prove that DCL could be adopted to any algorithms addressing this field.

V. Conclusion

In this paper, we design a training strategy imitating Curriculum Learning (CL) for crowd counting named Density-aware Curriculum Learning (DCL). It is not applied at the sample level like traditional CL, but at the pixel level for regression tasks. During the training process, DCL leverages ground truth to generate an attention map, giving more attention to these simple pixels defined by the density-aware curriculum. Besides, we propose a Pixel Shuffle Crowd Counter (PSCC) to verify and explore the availability of DCL. PSCC adopts VGG-16 as the encoder, and a lightweight decoder named Pixel Shuffle Decoder (PSD), which can express more density information without increasing the number of layers. Ultimately, it achieves competitive results in majority mainstream datasets while cooperating with DCL strategy. By the way, we also apply DCL on another fashion crowd counting models, and related experiments show that DCL can improve the performance of any crowd counter.

However, the setting of super parameters is not easy to determine. In the future, we will explore a better and reasonable way of determining their values, which can further reduce the difficulty of applying DCL, and enhance models getting higher generalization performance.

References

[1] A. S. Rao, J. Gubbi, S. Marusic, and M. Palaniswami, “Crowd event detection on optical flow manifolds,” *IEEE Transactions on Cybernetics*, vol. 46, no. 7, pp. 1524–1537, 2016.

[2] Y. Yuan, Y. Feng, and X. Lu, “Statistical hypothesis detector for abnormal event detection in crowded scenes,” *IEEE Transactions on Cybernetics*, vol. 47, no. 11, pp. 3597–3608, 2017.

[3] Y. Yuan, J. Fang, and Q. Wang, “Online anomaly detection in crowd scenes via structure analysis,” *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 548–561, 2015.

[4] V. J. Kok and C. S. Chan, “Grcs: Granular computing-based crowd segmentation,” *IEEE Transactions on Cybernetics*, vol. 47, no. 5, pp. 1157–1168, 2017.

[5] X. Li, M. Chen, F. Nie, and Q. Wang, “Locality adaptive discriminant analysis.” in *IJCAI*, 2017, pp. 2201–2207.

[6] W. K. Chow and C. M. Ng, “Waiting time in emergency evacuation of crowded public transport terminals,” *Safety Science*, vol. 46, no. 5, pp. 844–857, 2008.

[7] K. Al-Kodmany, “Crowd management and urban design: New scientific approaches,” *Urban Design International*, vol. 18, no. 4, pp. 282–295, 2013.

[8] P. R. De Almeida, L. S. Oliveira, A. S. Britto Jr, E. J. Silva Jr, and A. L. Koerich, “Pklot–a robust dataset for parking lot classification,” *Expert Systems with Applications*, vol. 42, no. 11, pp. 4937–4949, 2015.

[9] Q. Wang, J. Wan, and X. Li, “Robust hierarchical deep learning for vehicular management,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4148–4156, 2018.

[10] J. Shao, K. Kang, C. Change Loy, and X. Wang, “Deeply learned attributes for crowded scene understanding,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4657–4666.

[11] B. Zhou, X. Wang, and X. Tang, “Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2871–2878.

[12] C. Arteta, V. Lempitsky, and A. Zisserman, “Counting in the wild,” in *European conference on computer vision*. Springer, 2016, pp. 483–498.

[13] Y. Zhou, S. Huo, W. Xiang, C. Hou, and S. Kung, “Semi-supervised salient object detection using a linear feedback control system model,” *IEEE Transactions on Cybernetics*, vol. 49, no. 4, pp. 1173–1185, 2019.

[14] B. Leibe, E. Seemann, and B. Schiele, “Pedestrian detection in crowded scenes,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1. IEEE, 2005, pp. 878–885.

[15] W.-G. Chen, X. Wang, H.-Y. Wang, and H.-Y. Peng, “Hybrid approach using map-based estimation and class-specific hough forest for pedestrian counting and detection,” *IET Image Processing*, vol. 8, no. 12, pp. 771–781, 2014.

[16] F. Nie, Z. Wang, R. Wang, Z. Wang, and X. Li, “Towards robust discriminative projections learning via non-greedy $l_{2,1}$-norm minmax,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[17] Z. Wang, F. Nie, L. Tian, R. Wang, and X. Li, “Discriminative feature selection via a structured sparse subspace learning module,” in *Proc. Twenty-Ninth Int. Joint Conf. Artif. Intell.*, 2020, pp. 3009–3015.

[18] X. Li, M. Chen, F. Nie, and Q. Wang, “A multiview-based parameter free framework for group detection,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI’17. AAAI Press, 2017, p. 4147–4153.

[19] A. B. Chan and N. Vasconcelos, “Counting people with low-level features and bayesian regression,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2160–2177, 2011.

[20] K. Chen, C. C. Loy, S. Gong, and T. Xiang, “Feature mining for localised crowd counting.” in *BMVC*, vol. 1, no. 2, 2012, p. 3.

[21] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, “Single-image crowd counting via multi-column convolutional neural network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 589–597.

[22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations, ICLR 2015*, Y. Bengio and Y. LeCun, Eds., 2015.

[23] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[25] Y. Li, X. Zhang, and D. Chen, “Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1091–1100.

[26] Q. Wang, J. Gao, W. Lin, and Y. Yuan, “Pixel-wise crowd understanding via synthetic data,” *International Journal of Computer Vision*, pp. 1–21, 2020.

[27] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah, “Composition loss for counting, density map estimation and localization in dense crowds,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 532–546.

[28] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[29] Z. Wang, Z. Xiao, K. Xie, Q. Qiu, X. Zhen, and X. Cao, “In defense of single-column networks for crowd counting,” in *British Machine Vision Conference 2018, BMVC 2018*. BMVA Press, 2018, p. 78.

[30] D. Kang and A. B. Chan, “Crowd counting by adaptively fusing predictions from an image pyramid,” in *British Machine Vision Conference 2018, BMVC 2018*. BMVA Press, 2018, p. 89.

[31] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, “Crowdnet: A deep convolutional network for dense crowd counting,” in *Proceedings of the 24th ACM international conference on Multimedia*. ACM, 2016, pp. 640–644.

[32] D. B. Sam, S. Surya, and R. V. Babu, “Switching convolutional neural network for crowd counting,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 4031–4039.

[33] V. A. Sindagi and V. M. Patel, “Generating high-quality crowd density maps using contextual pyramid cnns,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1861–1870.

[34] M. Hossain, M. Hosseinzadeh, O. Chanda, and Y. Wang, “Crowd counting using scale-aware attention networks,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1280–1288.

[35] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, “Multi-scale convolutional neural networks for crowd counting,” in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 465–469.

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

[37] X. Cao, Z. Wang, Y. Zhao, and F. Su, “Scale aggregation network for accurate and efficient crowd counting,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750.

[38] D. B. Sam and R. V. Babu, “Top-down feedback for crowd counting convolutional neural network,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*. AAAI Press, 2018, pp. 7323–7330.

[39] D. Deb and J. Ventura, “An aggregated multicolumn dilated convolution network for perspective-free counting,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 195–204.

[40] J. Yang, Y. Zhou, and S.-Y. Kung, “Multi-scale generative adversarial networks for crowd counting,” in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 3244–3249.

[41] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, and H. Wu, “Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding,” *arXiv preprint arXiv:1811.11968*, 2018.

[42] J. Wan, N. S. Kumar, and A. B. Chan, “Fine-grained crowd counting,” *arXiv preprint arXiv:2007.06146*, 2020.

[43] J. Wan and A. Chan, “Adaptive density map generation for crowd counting,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1130–1139.

[44] Q. Wang, J. Gao, W. Lin, and X. Li, “Nwpu-crowd: A large-scale benchmark for crowd counting and localization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[45] J. Liu, C. Gao, D. Meng, and A. G. Hauptmann, “Decidenet: Counting varying density crowds through attention guided detection and density estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5197–5206.

[46] Y. Zhou, J. Yang, H. Li, T. Cao, and S. Kung, “Adversarial learning for multiscale crowd counting under complex scenes,” *IEEE Transactions on Cybernetics*, pp. 1–10, 2020.

[47] S. Li, X. Zhu, Q. Huang, H. Xu, and C. J. Kuo, “Multiple instance curriculum learning for weakly supervised object detection,” in *British Machine Vision Conference 2017, BMVC 2017*. BMVA Press, 2017.

[48] D. Zhang, J. Han, L. Zhao, and D. Meng, “Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework,” *International Journal of Computer Vision*, vol. 127, no. 4, pp. 363–380, 2019.

[49] D. Zhang, D. Meng, L. Zhao, and J. Han, “Bridging saliency detection to weakly supervised object detection based on self-paced curriculum learning,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, 2016, pp. 3538–3544.

[50] R. Lotfian and C. Busso, “Curriculum learning for speech emotion recognition from crowdsourced labels,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 4, pp. 815–826, 2019.

[51] Y. Wang, W. Gan, W. Wu, and J. Yan, “Dynamic curriculum learning for imbalanced data classification,” *arXiv preprint arXiv:1901.06783*, 2019.

[52] R. K. Vuddagiri, H. K. Vydana, and A. K. Vuppala, “Curriculum learning based approach for noise robust language identification using dnn with attention,” *Expert Systems with Applications*, vol. 110, pp. 290–297, 2018.

[53] N. Sarafianos, T. Giannakopoulos, C. Nikou, and I. A. Kakadiaris, “Curriculum learning for multi-task classification of visual attributes,” in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[54] M. R. U. Saputra, P. P. de Gusmao, S. Wang, A. Markham, N. Trigoni, Y. Almalioglu, M. Saputra, P. de Gusmao, A. Markham, N. Trigoni *et al.*, “Learning monocular visual odometry through geometry-aware curriculum learning,” in *IEEE International Conference on Robotics and Automation*, 2018, pp. 1–11.

[55] A. Surendranath and D. B. Jayagopi, “Curriculum learning for depth estimation with deep convolutional neural networks,” in *Proceedings of the 2Nd Mediterranean Conference on Pattern Recognition and Artificial Intelligence*. ACM, 2018, pp. 95–100.

[56] Q. Dong, S. Gong, and X. Zhu, “Multi-task curriculum transfer deep learning of clothing attributes,” in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 520–529.

[57] L. Gui, T. Baltrušaitis, and L.-P. Morency, “Curriculum learning for facial expression recognition,” in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 505–511.

[58] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, 2016, pp. 1874–1883.

[59] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[60] J. Gao, W. Lin, B. Zhao, D. Wang, C. Gao, and J. Wen, “C^3 framework: An open-source pytorch code for crowd counting,” *arXiv preprint arXiv:1907.02724*, 2019.

[61] Z. Cong, K. Kai, H. Li, X. Wang, X. Rong, and X. Yang, “Data-driven crowd understanding: A baseline for a large-scale crowd dataset,” *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1048–1061, 2016.

[62] C. Zhang, H. Li, X. Wang, and X. Yang, “Cross-scene crowd counting via deep convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 833–841.

[63] J. Gao, Q. Wang, and X. Li, “Pcc net: Perspective crowd counting via spatial convolutional network,” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2019.

[64] J. Gao, Q. Wang, and Y. Yuan, “SCAR: spatial-/channel-wise attention regression networks for crowd counting,” *CoRR*, vol. abs/1908.03716, 2019.

[65] M. Shi, Z. Yang, C. Xu, and Q. Chen, “Revisiting perspective information for efficient crowd counting,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[66] X. Wu, Y. Zheng, H. Ye, W. Hu, J. Yang, and L. He, “Adaptive scenario discovery for crowd counting,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, 2019, pp. 2382–2386.

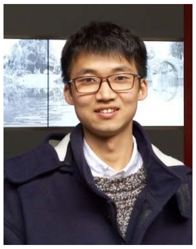
[67] X. Jiang, Z. Xiao, B. Zhang, X. Zhen, X. Cao, D. Doermann, and L. Shao, “Crowd counting and density estimation by trellis encoder-decoder networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[68] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, “Multi-source multi-scale counting in extremely dense crowd images,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2547–2554.

[69] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Qi Wang (M'15-SM'15) received the B.E. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. He is currently a Professor with the School of Computer Science, and with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.

Wei Lin received the B.E. degree in information security from the Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China, in 2018. He is currently pursuing the Master degree from Center for Optical Imagery Analysis and Learning, Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.



Junyu Gao received the B.E. degree in computer science and technology from the Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China, in 2015. He is currently pursuing the Ph.D. degree from Center for Optical Imagery Analysis and Learning, Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.

Xuelong Li (M'02-SM'07-F'12)2) is currently a Full Professor with the School of Computer Science and the Center for OPTical Imagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University,Xi'an 710072, China.