Multiscale Multitask Deep NetVLAD for Crowd Counting

Zenglin Shi, Le Zhang, Yibo Sun, and Yangdong Ye

***Abstract*—Deep convolutional networks (CNNs) reign undisputed as the new de-facto method for computer vision tasks owning to their success in visual recognition task on still images. However, their adaptations to crowd counting have not clearly established their superiority over shallow models. Existing CNNs turn out to be self-limiting in challenging scenarios such as camera illumination changing, partial occlusions, diverse crowd distributions, and perspective distortions for crowd counting because of their shallow structure. In this paper, we introduce a dynamic augmentation technique to train a much deeper CNN for crowd counting. In order to decrease overfitting caused by limited number of training samples, multitask learning is further employed to learn generalizable representations across similar domains. We also propose to aggregate multiscale convolutional features extracted from the entire image into a compact single vector representation amenable to efficient and accurate counting by way of "Vector of Locally Aggregated Descriptors" (VLAD). The "deeply supervised" strategy is employed to provide additional supervision signal for bottom layers for further performance improvement. Experimental results on three benchmark crowd datasets show that our method achieves better performance than the existing methods. Our implementation will be released at https://github.com/shizenglin/Multitask-Multiscale-Deep-NetVLAD.**

***Index Terms*—Crowd counting, deep learning, multitask learning, vector of locally aggregated descriptors (VLAD).**

I. Introduction

CROWD counting refers to estimating the number of pedestrians in the still images or videos from surveillance cameras [1]. It has been an active research topic in computer vision due to its wide ranging applications including surveillance, resource management, urban planning, abnormality detection [2], and commercial profit [3]. Good methods of crowd counting can also be extended to other domains, for instance, counting cells or bacteria from microscopic images, animal crowd estimates in wildlife sanctuaries, or estimating the number of vehicles at transportation hubs or traffic jams, etc.

Basically, existing methods for crowd counting can be divided into two categories: detection models and regression models [4]. Detection models work by detecting individual object in crowded scenarios with pretrained detectors [5]. They usually work well in low-density scenes and may not generalize to challenging scenarios such as occlusions, complex background, and congested scenes because of detection failure [6]. Regression methods, on the other hand, are to directly regress pedestrians' number through a regression function with some visual descriptors such as texture feature, edge features [1], [3], [4], or learned representations [7], [8]. These early methods have consistently improved the counting performance leading the research community to address more challenging scenarios and complex datasets [7], [8]. However, significant hurdles due to camera illumination changing, partial occlusions, diverse crowd distributions, and perspective distortions render existing approaches practically obsolete.

Recently, deep CNNs have made vast inroads into computer vision due to their commendable performance [9]–[13]. Motivated by this, CNNs for crowd counting [7], [8], [14]–[18] have been proposed with state-of-the-art results on some crowd counting benchmarks. In [7], a network was fine-tuned on similar datasets from same domain to enhance the overall performance of the counting system. Wang *et al.* [14] used a CNN-based network to count pedestrians in the high dense crowd by directly regressing the number of pedestrians. Boominathan *et al.* [16] proposed a framework consisting of both deep and shallow networks to get crowd density map. The proposed system are reported to be more robust with perspectives and scale variations. Zhang *et al.* [8] proposed a multicolumn CNN architecture that consists of filters of different sizes to predict the crowd. "Hydra CNN" [17] was proposed to estimate object densities in different very crowded scenarios in a scale-aware manner. A single column fully convolutional crowd counting model was introduced in [15]. Based on the multiscale blobs, Zeng *et al.* [18] introduced a multiscale crowd model to generate scale-relevant features for higher crowd counting performance in a single-column architecture.

On one hand, all the aforementioned approaches worked on a relative shallow CNN model. Modifying them to a much deeper network structure to further boost the discriminative ability of the learned representations for crowd counting is not straightforward due to limited training data. This motivates us to study

Manuscript received February 22, 2018; revised April 26, 2018; accepted May 25, 2018. Date of publication July 2, 2018; date of current version November 1, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61772475 and the National Key R&D Plan under Grant 2018YFB1201403. Paper no. TII-18-0494. *(Corresponding author: Yangdong Ye.)*

Z. Shi, Y. Sun, and Y. Ye are with the School of Information Engineering, Zhengzhou University, Zhengzhou 450066, China (e-mail: iezlshi@gs.zzu.edu.cn; ieybsun@gs.zzu.edu.cn; ieydye@zzu.edu.cn).

L. Zhang is with the Advanced Digital Sciences Center, University of Illinois at Urbana-Champaign, Singapore 138632 (e-mail: zhang.le@adsc.com.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2018.2852481

the problem of training deep CNNs methods on existing datasets with less risk of overfitting. To address this, we first introduce novel data augmentation techniques which can generate diverse training samples and prevent severe overfitting. Moreover, in order to alleviate overfitting caused by scarceness of large-scale well-annotated dataset, multitask learning [19] is employed to learn generalizable representations across different datasets. On the other hand, existing CNNs based approaches usually use features from the highest layers which is far from satisfactory in the sense of employing rich hierarchies from different layer of CNNs. As reported in [10] and [20], different layers encode different types of features. Higher layers capture semantic concepts on object categories, whereas lower layers encode more discriminative features to capture intraclass variations. As another contribution, we shed light on how to learn generalizable features for crowd counting by aggregate multiscale convolutional features extracted from the entire image into a compact single vector representation by way of VLAD [21], [22]. The "deeply supervised" strategy is employed in the bottom layers to provide additional supervision signal toward better performances. In this way, both features from top layer and intermediate layers, which may correspond to objects with smaller and larger scales, respectively, can automatically learn to contribute in the final counting results. The proposed system is end-to-end trainable and is more effective and robust with scale variation, viewpoint changes and occlusions in congested scenes. We evaluate our method on three benchmark datasets including the challenging UCF_CC_50 dataset [23], Shanghaitech dataset [8], and WorldExpo'10 dataset [7]. Experimental results demonstrate our method achieves excellent performance compared to the existing methods. To summarize, we make the following contributions:

1) A principled way of combining similar datasets based on multitask learning is employed to address the problem of limited training samples.
2) We propose to aggregate multiscale convolutional features extracted from the entire image into a compact single vector representation by way of VLAD. The "deeply supervised" strategy is employed to provide additional supervision signal for bottom layers toward better representation learning.
3) The proposed system is end-to-end trainable and outperforms state-of-the-art methods on several benchmarking datasets.

II. Related Work

The wealth of research in this area is such that we cannot give an exhaustive review. Instead, we will focus on describing the most important threads of relevant work.

A. Convolutional Neural Networks for Crowd Counting

In recent years, CNNs have become ubiquitous in computer vision owing to their success in visual recognition tasks on still images success in large-scale visual recognition tasks. Learned representation with a data-driven manner in CNNs usually outperforms hand-crafted low-level features [10]. Building on the lessons learnt from the current astonishing performances, lots of CNN-based models such as AlexNet [9], VGGNet [12], and GoogleNet [11] have been developed to solve various computer vision tasks including crowd counting. Based on VGGNet, Boominathan *et al.* [16] proposed a framework consisting of both deep and shallow networks to get the density map. However, most CNN-based crowd models were developed as shallow models because of limited training data [7], [8], [14], [15], [17], [18]. Single column shallow CNN fails to predict the crowd count when facing complex scene backgrounds and scale variations problems [7]. Multicolumn structures remedy this problem to some degree [8]. However, they are usually much more difficult to train because each subnet may have different loss surfaces. Our method bears some similarity to the work of [7], where a similar deep VGG network is employed. However, several distinctive designs such as novel data augmentation techniques, multiscale and multitask learning makes the proposed method performs much better than [7]. More specifically, Zhang *et al.* [7] works in a synergistic manner where a shallow network is employed to assist the deep model. In the proposed method, both high-level and low-level features are encapsulated into a single network allowing end-to-end "deeply supervised" learning. Moreover, novel dynamic data augmentation techniques as well as multitask learning are employed to further decrease overfitting caused by limited training samples.

B. Multiscale Features for Crowd Counting

One of the most challenging part of crowd counting comes from the significant variations in pedestrian scale, which is usually induced by significant variation along the depth direction. This has been previously addressed by adopting perspective maps to rescale pedestrian into the same size [7]. However, perspective map may fail to work and even bring undesirable distortions when there exist significant overlaps between people. Alternatively, this can be tackled by inserting different receptive filed into the system. For example, CrowdNet [16] employs both deep and shallow networks where each network has different receptive field and can count well for different scales. Zhang *et al.* [8] further extends this idea to a multicolumn setting to have more than two networks. Compared with existing works which fuse multiple networks, the proposed method is much more efficient in a sense that features from different layers, which corresponds to different scales, are fully optimized. It enhance scale robustness in a single network via the well-established "deeply supervised" strategy [24], [25]. More specifically, for bottom layers, apart from supervision signals back-propagated from top layers, we provide additional direct supervisions based on the groundtruth. In this way, features from both bottom layers and top layers can learn to focus on pedestrian being with large and small scales, respectively.

C. Texture Information

Texture usually refers to the spatial organization of a set of basic elements or primitives (i.e., textons), the fundamental microstructures in natural images and the atoms of preattentive human visual perception [26]. Unlike image categorization which

TABLE I
EFFECTIVENESS OF TEXTURE FEATURES FOR CROWD COUNTING

Method	MAE	RMSE
dense SIFT+VLAD+GPR	522.2	**659.8**
VGG16	**518.1**	677.7

seeks for global information on the categories, crowd counting, as illustrated in Fig. 3, could be even more challenging and local textures may play an important role. This has been previously studied in [23] and [27]–[29]. In order to further verify this, we have done two quantitative experiments on UCF_CC_50 dataset. Dense SIFT features, which have been widely applied to capture texture information in image registration and object recognition, are extracted from the patches of all the images. Afterward, the features are encoded using k-means with a cluster size of 256. Next, *Vector of Locally Aggregated Descriptors (VLAD)* [30], which works extremely well for extracting texture information and will be introduced later, are used to generate the final representation. Finally, a *Gaussian process (GP)* [31] is used to regress feature vectors to the number of people per image patch. Surprisingly, we found that this simple method performs competitively against fully optimized deep VGG-16 networks, as demonstrated in Table I.

This motivates us to investigate the usage of VLAD, which has been reported to be successful in encoding the texture information from image [22], [32]. VLAD learns a set of vector representations of an image by aggregating descriptors based on a locality criterion in feature space. VLAD usually works on discriminative local descriptors such as SIFT [33]. It can be viewed as a simplified version of *Fisher Vector* image descriptor [34]. VLAD bears some similarities with bag-of-visual-words [35] by computing data statistics on top of local descriptors.

Given N local image descriptors x_i as input, learning a codebook $C = \{c_1, ...c_k\}$ of K visual words with k-means, assuming the local descriptor to be D-dimensional, the output VLAD image representation V is $K \times D$-dimensional. Using V as a $K \times D$ matrix, the (j, k) element of V is computed as follows:

$$V(j,k) = \sum_{i=1}^{N} a_k(X_i)(x_i(j) - c_k(j)) \tag{1}$$

where $x_i(j)$ and $c_k(j)$ respectively denote the jth component of the descriptor x_i considered and of its corresponding visual word c_k. $a_k(x_i)$ denotes the membership of the descriptor x_i to kth visual word, i.e., it is 1 if cluster c_k is the closest cluster to descriptor x_i and 0 otherwise. For each of the k clusters, the residuals $(x_i - c_k)$ are accumulated, and the sum of residuals of descriptors are assigned to cluster c_k, being each D-dimensional column k of V. The matrix V is subsequently L2-normalized by $V := V/||V||_2$.

In this way, VLAD generates a fixed-size compact description of a set comprising a variable number of data points. VLAD has been reported to perform well in many computer vision tasks such as image retrieval and recognition because of its robustness to occlusion and viewpoint changes [36].

To mimic VLAD in a CNN framework, Arandjelovic *et al.* [22] designed a trainable generalized VLAD layer, NetVLAD. The feature captured by NetVLAD is a powerful image representation trainable end-to-end on the target task. To make the layer amenable to training via backpropagation, the NetVLAD layer plugs a soft-assignment into the original VLAD descriptor. The (j, k) element of the resulting VLAD descriptor V is computed as follows:

$$V(j,k) = \sum_{i=1}^{N} \frac{e^{w_k^T x_i + b_k}}{\sum_{k'} e^{w_{k'}^T x_i + b_{k'}}} (x_i(j) - c_k(j)) \tag{2}$$

where $x_i(j)$ and $c_k(j)$ are the jth dimensions of the ith descriptor and kth cluster center, respectively. w_k, b_k, and c_k are sets of trainable parameters for each cluster k. We use the NetVLAD layer on the top of concatenation layer which fuses multilevel features from VGG-16, which helps us to obtain significant robustness to translation, partial occlusion, and pedestrian scale. Following [22], we use $K = 64$ in our experiments.

III. MULTISCALE MULTITASK DEEP NETVLAD CROWD MODEL

Crowd images are always captured from different scenes under challenging scenarios such as scale variations, perspective distortion, and partial occlusions. Existing CNN-based crowd models usually fail to learn generalizable features for those cases because of their shallow structure. In this paper, we introduce a much deeper multiscale multitask model based on VGG-16 network [12]. An overview of our architecture is shown in Fig. 1.

A. Deep NetVLAD Model

Our network is slightly different from standard network used in [12]. The VGG network in [12] was originally proposed for object classification. Unlike conventional CNN network, VGG-16 network generalizes well by simply stacking convolution filters with a fixed kernel size of 3×3. Stacking multiple 3×3 convolution filters is reported to be as effective as, or usually superior than, adopting larger receptive filed such as 7×7 or 5×5. VGG-16 network usually generalizes well to other vision tasks including crowd counting [16] owing to its excellent feature learning ability. The original VGG network has five max-pool layers each with a stride of two and, hence, the resultant output features have a spatial resolution of only 1/32 times the input image. However, empirical studies find that such a low-resolution feature output turns out to be lacking for crowd counting. In our adaptation of the VGG model, we set the stride of the fourth max-pool layer to 1 and remove the fifth pooling layer. To handle the receptive-field mismatch caused by the removal of stride in the fourth max-pool layer, we double the receptive field of convolutional layers after the fourth max-pool layer by using dilated convolution [37], thereby enabling them to operate with their originally trained receptive field. This gives us a much larger feature map with the network to make predictions at one-eighth times the input resolution.

Feature maps from the last convolutional layer capture higher conceptual information about the object exists in the image and,

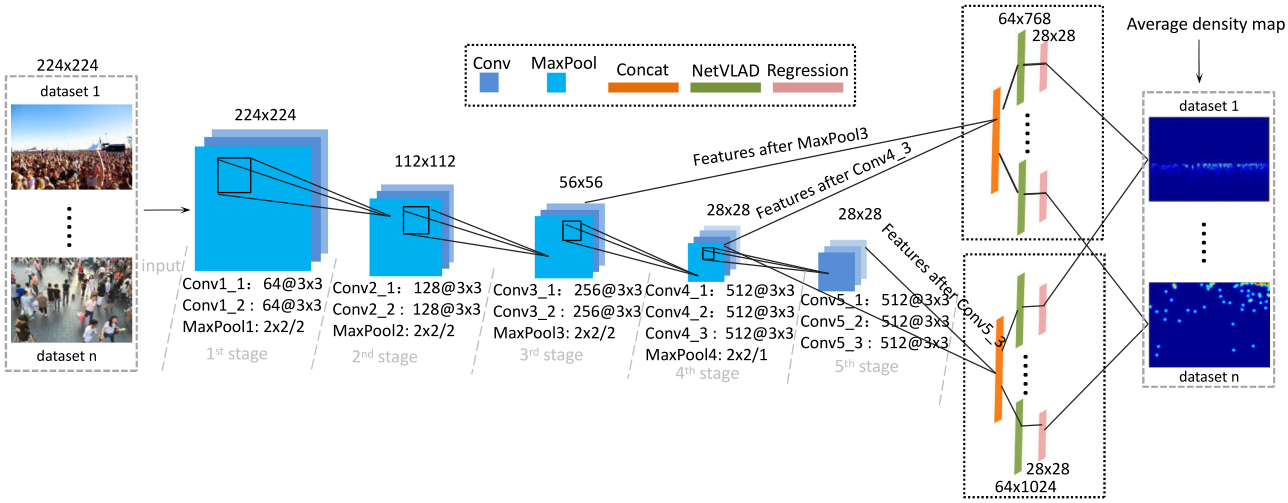


Fig. 1. Architecture of multiscale multitask deep NetVLAD crowd model. A VGG-like network is adopted here. The features learned by convolutional layers in third and fourth stage are concatenated first and further processed by a NetVLAD layer to generate a fixed-size compact description. Similarly, the features learned by convolutional layers in fourth and fifth stage are also concatenated and fed into another NetVLAD layer. Then, the VLAD description with intranormalization is fed to the regression layer to obtain the predicted density map. Each regression layer after NetVLAD is supervised by the ground truth. Averaging the output of these two regression layer obtains the final predicted density map. The whole network is trained with multitask learning. It receives a combined dataset and each branch of the system, dedicated to its own dataset, is responsible for estimating the density map therein. More details can be found in Section III-D.

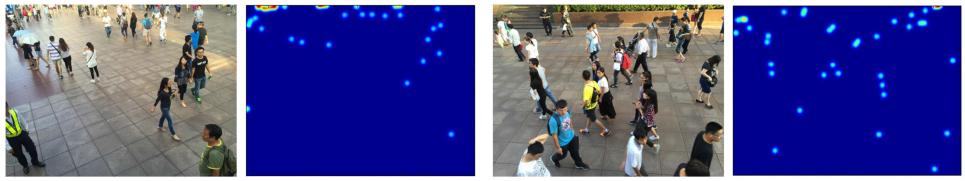
thus, can be used to regress the object pedestrains' number in the crowd scene, as done in [16]. However, due to challenging factors such as camera illumination changing, partial occlusions, diverse crowd distributions, and perspective distortions, conventional approach used convolutional layer turns out to overfit severely in practice. This motivates us to study some alternative strategy which can bring certain level of invariance under such scenarios. To address this, we adopt NetVLAD [22], which is readily pluggable into any existing deep architecture and enable the whole system end-to-end trainable. NetVLAD was reported to perform better than conventional convolutional layer in several vision tasks such as place recognition [22] and image retrieval[38]. It is less prone to overfitting by aggregating information about the statistics of local descriptors aggregated over the image.

B. Scale Invariance

Existing CNNs for crowd counting usually regress the number of object from the highest layer of feature maps which turns out to be lacking in practice. Actually deep CNNs may generate rich feature hierarchies from input and how to transfer them in an effectively manner for crowd counting remains to be an open question. As reported in [10] and [20], different layers encode different types of features. Higher layers capture semantic concepts on object categories, whereas lower layers encode more discriminative features to capture intraclass variations. We empirically found features from lower layer containing rich information for people counting. In order to show this, we conduct an experiment using features from lower layer for crowd counting. In order to gain some invariance in different input scales, here we propose to fuse both highest and lower level features from deep CNNs by delivering their concatenation to NetVLAD layer. In this way, interactions of feature maps from different layer with complementary information are effectively considered and can be jointly optimized by efficient backpropagation. This bears some similarity in [16] which employs a combination of both deep and shallow networks. However, our solution turns out to be computationally more efficient as only one single feed forward is needed to count. This is more attractive in an resource-constrained scenario. It also allows richer feature image representations to be generated which plays a central role in outperforming existing state-of-the-art method. In addition, we take inspirations from well-established "deeply supervised" strategy [24], [25] to further optimize intermediate layers which could work better for pedestrian beings with larger scales. For those layers, apart from supervision signals backpropagated from top layers, we provide additional direct supervisions based on the groundtruth. In this way, features from both bottom layers and top layers can automatically learns to contribute in the final counting results.

C. Data Augmentation

Unlike image classification [12], deep CNNs to crowd counting have not clearly established their superiority over shallower structures. First, we believe that lack of large-scale image data hampers the efforts to adopt much deeper CNNs for crowd counting. For example, the largest crowd counting data are Shanghaitech dataset, which contains only 3980 images and is much smaller compared with Imagenet [9]. However, deep CNNs always require a huge number of training samples to tune the network weights. Moreover, the task of crowd counting is more complex than identifying the category of image or detect salient object therein. Introducing large-scale datasets for crowd counting may partially alleviate the problem. However, manual labeling is costly, time-consuming, and error prone. It can also raise privacy concerns. It is often impractical as there



(a) (b)

Fig. 2. Two example frames with the same number of pedestrains and the different distribution of pedestrains.

may exist several thousands of people within a single image in dense crowd scenarios. To address this problem, we study several good data augmentation techniques for effectively training of deep CNNs and reduce the effect of overfitting. By carefully training our proposed deep CNNs on existing dataset with these data augmentation strategies, we are able to achieve the state-of-the-art performance on several challenging benchmark datasets.

It is nontrivial to apply off-the-shelf data augmentation techniques in public available deep learning framework. Consider conventional data augmentation techniques for image categorizing, one may only need to augment the input images without touching its target. This is because high-level concept of the image of the augmented images, such as their categorizations, would not change. However, in crowd counting, augmentations should be performed for both input and ground-truth images. For example, when one image patch is scaled down, its density map should also be scaled and most importantly, the total number of people therein should remain the same.

Our data augmentation methods differ with conventional crowd counting system in the sense that we perform it dynamically. Conventional methods first enlarge existing dataset by applying predefined augmentation techniques in an offline manner. Then, the same extended data are applied in each epoch during the training process. For each iteration, we randomly generates training data by randomly flipping, randomly changing image scale, and randomly cropping image patches in different location during training. In this way, these strategies helps us to generate a much larger if not infinite and diversified training samples in a randomized manner which is crucial in reduce overfitting.

D. Multitask Learning

Apart from effective data augmentation, one could consider combining different datasets into one to decrease overfitting. To this end, we consider a well-established method called multitask learning [19], [39]. It aims at learning a generalizable representation, which is applicable not only to the task in question, but also to other tasks with significant commonalities. In the proposed method, since a shared network backbone is employed by all tasks, additional tasks act as a regularization which requires the system to perform well on a related task. The backpropagation training from different tasks will directly impact the representation learning of shared parameters. It prevents overfitting by solving all tasks jointly and allowing for the exploitation of additional training data.

In our case, the CNN architecture is modified to have several branches after concatenation layer as shown in Fig. 1: each branch, equipped with its own loss and operated only on the images coming from the respective dataset, is responsible for estimating the density map therein. More specifically, we categorize different datasets into two scenarios. The first one, consisting of Shanghaitech part B and WorldExpo'10 dataset, considers the case where the pedestrian is relatively sparse. The other one, making up of UCF_CC_50 and Shanghaitech part A dataset, addresses the problem where the number of pedestrians is much more larger. Each of these two scenarios can be effectively solved by multitask learning. The parameters of each branch as well as the CNN backbone can be jointly optimized by backpropagation.

E. Crowd Density Map

An straightforward approach to count is to regress the actual total head counts within an image. This may turn out to be inaccurate in practice because head counts preserves too little crowd information. An alternative is to regress a density map created based on annotated pedestrians's spatial location [40] instead of a single number. The final pedestrians' number is obtained by integration on the density map. An density map contains much richer and much detailed information on local regions. For example, in Fig. 2, the number of pedestrains in those two images is the same. However, the distribution of pedestrains differs significantly and the CNNs can learn to distinguish different patterns by regress on density map. It also makes CNNs more sensitive to pedestrians of different sizes and perspective variation.

Creating a high-quality density map as the ground truth is crucial to train an accurate crowd CNN-based model. For a training image I_i, the ground-truth density function can be defined as a kernel density estimate based on the annotated pedestrians's points

$$F(p) = \sum_{p \in P} \mathcal{N}(p, P, \sigma) \tag{3}$$

where p denotes a pixel, $\mathcal{N}(p, P, \sigma)$ denotes a normalized 2-D Gaussian kernel evaluated at p, with the mean at the user-placed dot P, and an isotropic covariance matrix with σ being a small value.

The spread parameter σ should be determined reasonably. Motivated by [7], the σ is set to $0.2M(p)$ usually when the perspective map $M(p)$ can be available. However, generating such perspective maps is a laborious task and involves intensively manually labeling which turns out to be unpractical. Zhang *et al.* [8] provide an alternative strategy to determine σ in an adaptive manner. The spread parameter σ is determined based

TABLE II
COMPARISON OF THREE BENCHMARK DATASETS IN THE NUMBER OF IMAGES (NUM), THE MAXIMAL CROWD COUNT (MAX), THE MINIMAL CROWD COUNT (MIN), THE AVERAGE CROWD COUNT (AVE), AND TOTAL NUMBER OF LABELED PEDESTRIANS (TOTAL)

Dataset		Resolution	Num	Max	Min	Ave	Total
UCF_CC_50		various	50	4543	94	1279.5	63,974
WorldExpo'10		576×720	3980	253	1	50.2	199,923
Shanghaitech	Part_A	various	482	3139	33	501.4	241,677
	Part_B	768×1024	716	578	9	123.6	88,488

Fig. 3. Some example frames of three benchmark datasets: UCF_CC_50 dataset, Shanghaitech dataset, and WorldExpo'10 dataset.

on the size of the head for each person within the image. However, it is difficult to obtain the size of the head due to commonly happened occlusions. In this case, an assumption is made that the crowd is somewhat evenly distributed around each head. Then, the average distance $\bar{d}$ between the head and its nearest k neighbors (in the image) give a reasonable estimate of head size. As a result, σ can be set to $\beta\bar{d}$. Empirically, β is suggested to be 0.3 to give the best result.

IV. EXPERIMENTS

We evaluate the proposed methods on three benchmark datasets: UCF_CC_50 dataset [23], Shanghaitech dataset [8], and WorldExpo'10 dataset [7]. The comparison and some example frames of these three benchmark datasets can be found in Table II and Fig. 3, respectively. Network structure used for this study is described in Fig. 1. Our crowd model regresses pedestrians' number through density map which is elaborated in Section III-E. Euclidean distance is used to guide the training process by calculating the difference between the output feature map and the corresponding ground truth. Equation (4) gives the loss function that needs to be optimized, where Θ represents the parameters of the model, and $F(X_i;\Theta)$ denotes the output of the model. X_i and F_i are respectively the ith input image and density map ground truth

$$L(\Theta) = \frac{1}{2N} \cdot \sum_{i=1}^{N} \parallel F(X_i;\Theta) - F_i \parallel_2^2 . \tag{4}$$

We implement our system in Matconvnet [41] on a single machine with a TitanX GPU. The networks are trained using *Stochastic Gradient Descent* with minibatch size of 10 at a fixed constant momentum value of 0.9. Momentum term is used for faster convergence. Weight decay with a fixed value of 0.0005 is used as a regularizer. We use a fixed learning rate of 0.0001 in the top five layer of our crowd model to enlarge the gradient signal for effectively parameter updating and use a relatively smaller learning rate of 0.00001 in other layer of our crowd model.

A. Evaluation Metric

The widely used *mean absolute error* (MAE) and the *root mean squared error* (RMSE) are adopted to evaluate the performance of different methods. The MAE and RMSE are defined as follows:

$$\text{MAE} = \frac{1}{N} \cdot \sum_{i=1}^{N} |(z_i - \tilde{z}_i)| \tag{5}$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} (z_i - \tilde{z}_i)^2}. \tag{6}$$

Here, N represents the total number of images in the testing datasets, z_i and $\tilde{z}_i$ are the ground truth and the estimated value, respectively, for the ith image. In general, MAE and RMSE indicate the accuracy and robustness of a method, respectively.

B. UCF_CC_50 Dataset

The challenging UCF_CC_50 dataset [23] contains 50 images that are randomly collected from Internet. The number of head ranges from 94 to 4543 with an average of 1280 individuals per image. The total number of annotated persons within 50 images is 63 974. Challenging issues such as large variations in head number among different images coming from small number of training images comes in the way of accurately counting for UCF_CC_50 dataset. We follow the standard evaluation protocol by splitting the dataset randomly into five parts in which each part contains ten images. Fivefold cross validation is employed to evaluate the performance. Since the perspective maps are not provided, we generate the ground truth density map by using the Zhang's method [8] as described in Section III-E.

We compared our method on this dataset with nine state-of-the-art methods. The work of[5], [23], and [40] used hand-crafted features such as HOG and dense SIFT to regress the density map. The work of [7] and [15] predicted the density map based on a single column shallow CNN crowd model. Three CNN-based methods in [8], [16], and [17] developed the multicolumn network to estimate the crowd count of an image. Based on the multiscale blobs, Zeng *et al.* [18] introduced a multiscale crowd model. Table III summarizes the detailed results. We can intuitively see that

1) our single-column, deep model outperforms others on counting accuracy;

TABLE III
COMPARING RESULTS OF DIFFERENT METHODS ON THE UCF_CC_50 DATASET

Method	MAE	RMSE
Rodriguez et al.[5]	655.7	697.8
Lempitsky et al.[40]	493.4	487.1
Isrees et al.[23]	419.5	541.6
Zhang et al. [7]	467.0	498.5
CrowdNet [16]	452.5	-
Zhang et al. [8]	377.6	509.1
Zeng et al. [18]	363.7	468.4
Mark et al. [15]	338.6	424.5
Daniel et al. [17]	333.7	425.2
NetVLAD	317.4	408.2
NetVLAD+MultiTask	**311.3**	**401.8**

TABLE IV
COMPARING PERFORMANCES OF DIFFERENT METHODS ON SHANGHAITECH DATASET

Method	Part_A		Part_B	
	MAE	RMSE	MAE	RMSE
LBR+RR	303.2	371.0	59.1	81.7
Zhang et al. [7]	181.8	277.7	32.0	49.8
Zhang et al. [8]	110.2	173.2	26.4	41.3
NetVLAD	109.4	175.9	22.7	35.4
NetVLAD+MultiTask	**107.6**	**169.3**	**21.4**	**33.9**

2) all of CNN-based crowd models have a better performance than handcraft features based methods;
3) compared with the single-column models, multicolumn, or multiscale models further boost the performance;
4) multitask learning improves our deep architecture by exploiting additional datasets from other tasks.

C. Shanghaitech Dataset

The Shanghaitech dataset [8] is a large-scale crowd counting dataset, which contains 1198 annotated images with a total of 330 165 persons. This dataset is the largest one in the literature in terms of the number of annotated pedestrians. It consists of two parts: Part_A consisting of 482 images are randomly captured from the Internet, and Part_B including 716 images are taken from the busy streets in Shanghai. Each part is divided into training and testing subset. The crowd density varies significantly among the subsets, making it difficult to estimate the number of pedestrians.

We compare our method with three existing methods on the ShanghaiTech dataset. The LBP+RR method used LBP feature to regress the function between the counting value and the input image. Zhang *et al.* [7] designed a convolutional network (CNN) to regress both the density map and the crowd count value from original pixels. A multicolumn CNN [8] is proposed to estimate the crowd count value and crowd density map. All the detailed results for each methods are illustrated in Table IV. First, it is obvious that all deep learning methods outperform hand-crafted features significantly. The shallow model in [8] employs a much wider structure by a multicolumn design performs better than the shallower CNN models in [7] in both cases. Our much deeper CNN outperforms shallower model in [7] in both cases. Multitask learning decreases the estimation error in each case, which further verifies the feasibility of combining similar dataset for training deep neural network for crowd counting. It is also interesting to see that our deeper structure performs competitively with wider structure on this dataset.

D. WorldExpo'10 Dataset

The WorldExpo'10 dataset [7] is a large-scale and cross-scene crowd counting dataset. It contains 1132 annotated sequences which are captured by 108 independent cameras, all from Shanghai 2010 WorldExpo'10. This dataset consists of 3980 frames with a total of 199 923 labeled pedestrians, which are annotated at the centers of their heads. Five different regions of interest (ROI) and the perspective maps are provided for the test scenes.

To be consistent with the previous works of [7], [8], we generate the labeled density map using perspective map as elaborated in Section III-E. Following the work of [8], we evaluate our method on the whole area of ROI. Thus, we set the area out of ROI to zero through ROI mask for both the labeled density map and the predicted density map. Table V shows the results of our method and other methods on this dataset. We observe that learned representations are more robust than the handcraft features. Our method outperforms all other in terms of average MAE. More specifically, our method without multitask learning achieves the best performance in two out of five scenes while method in [8] win two in other three cases, respectively. Employing multitask learning further wins 1 case and decreases the overall MAE by 0.3.

E. Ablation Study

To further investigate the effectiveness of different components of the proposed crowd counting system, we report the performance of different variants of the proposed algorithm on the hardest of the fivefolds from UCF_CC_50 dataset as [16] in this section. Similar results can be found for other two datasets.

1) Benefits of NetVLAD: To further understand the merits of NetVLAD, we conduct an experiment to demonstrate them. NetVLAD layer is replaced with a convolutional layer. This results in a fully convolutional network (FCN). FCNs can produce a proportionally sized output feature map for a given input image rather than a classification label or regression score. FCNs have been used for a variety of tasks including semantic segmentation [42], saliency prediction [43] as well as crowd counting task [8], [15], [16]. FCNs can also be beneficial for crowd counting because it can have variable input size, which reduce the loss of image detail and visual distortions typically encountered during image downsampling and reshaping. Existing results in the literature have shown superior performance for crowd counting in FCNs. In our experiment, motivated by [16], we replace the NetVLAD layer with 1×1 convolution layer to generate the predicted density map. For a fair comparison, we only deliver the last convolutional feature to the NetVLAD layer because FCN obtains the predicted density map just by

TABLE V
COMPARING RESULTS OF THE PROPOSED METHOD AND OTHERS ON MAE METRIC OF THE WORLDEXPO'10 DATASET

Method	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Average
LBP+RR	13.6	58.9	37.1	21.8	23.4	31.0
Zhang et al.[7]	9.8	14.1	14.3	22.2	3.7	12.9
Zhang et al. [8]	**3.4**	20.6	12.9	**13.0**	8.1	11.6
NetVLAD	4.1	**13.7**	12.5	17.2	**5.9**	10.6
NetVLAD+MultiTask	3.7	15.9	**10.2**	15.2	6.7	**10.3**

TABLE VI
RESULTS OF USING DIFFERENT LAYER ON THE UCF_CC_50 DATASET

Method	MAE	RMSE
dense SIFT+VLAD+GPR	522.2	659.8
fully convolution layer	518.1	677.7
NetVLAD layer	**450.8**	**614.6**

TABLE VII
RESULTS OF USING DIFFERENT FEATURES ON THE UCF_CC_50 DATASET

Method	MAE	RMSE
lower-level features	616.3	700.5
highest-level features	450.8	614.6
concatenation of lower and highest level features	409.1	586.7
multiscale	**396.3**	**564.2**

TABLE VIII
RESULTS OF USING DIFFERENT DATA AUGMENTATION STRATEGIES ON THE UCF_CC_50 DATASET

Method	MAE	RMSE
without augmentation	543.6	687.9
static augmentation	457.4	623.6
dynamic augmentation	**396.3**	**564.2**

using single-stage convolutional feature. As shown in Table VI, our system outperforms FCN in terms of both MSE and RMSE. We hypothesize that aggregating information about the statistics of local descriptors aggregated over the image, which can be handled by NetVLAD layer in our system, can be beneficial for crowd counting.

2) Superiority Over Hand-Crafted Features: By employing well-studied NetVLAD, our system is end-to-end trainable and can outperform hand-crafted features. In order to show this, we use a dense version of SIFT named dense SIFT as hand-crafted feature, which have been widely applied to image registration and object recognition [44]. To ensure a sufficient number of data for training, we perform data augmentation by cropping nine patches from each image and flipping them. We simply fix the nine cropped points as top, center, and bottom combining with left, center, and right. Each patch is 25% of the original size. Dense SIFT features are extracted from the patches of all the images using vl_phow [45]. Afterward, the features are encoded using k-means with a cluster size of 256. Next, VLAD are used to generate the final representation. Finally, a GP [31] is used to regress feature vectors to the number of people per image patch. The classes of functions that the GP can model is dependent on the kernel function used. Following [27], we combine the linear and the squared-exponential (RBF) kernels. As shown in Table VI, our NetVLAD layer achieves higher accuracy, as expected. It is also interesting to see that dense SIFT VLAD performs competitively against FCN.

3) Benefits of Multiscale Features: In deep CNN, higher layers capture semantic concepts on object categories, whereas lower layers encode more discriminative features to capture intra-class variations. However, this problem has never been investigated for crowd counting. We conduct an experiment to verify this. As shown in Fig. 1, the convolution output from the fifth stage and fourth stage of our crowd model is used as highest and lower level features, respectively. The lower level features, highest level features, and their concatenation are separately fed to the NetVLAD layer for adequate comparison. The detailed results can be found in Table VII. We can see that highest level features obtain better result than lower level features. However, we argue that lower level features could also provide complementary information for the same task. With different receptive field, features of different levels provide complementary information with different resolution about the same pedestrian. Interactions between features from different layers are well encapsulated in the proposed methods; hence, resulting in a compact single vector representation amenable to more accurate counting as shown in the last row of Table VII.

4) Importance of Data Augmentation: To mitigate the lack of sufficient training data, we study several good data augmentation strategies to generate more training data. We empirically find that horizontally flipping, randomly scaling, and randomly cropping are good strategies for crowd counting. They works to generate a much larger and more diversified dataset. However, under the constrain from hardware resources such as disk space and memory space, the usual way to do this is to generate fixed training data with limited predefined augmentation technique, i.e., using the same data when training deep model each epoch as the other counting methods do [7], [8], [16]. In this paper, we propose to augment data on the fly during the training process, i.e., randomly generating training data by randomly flipping, randomly changing image scale, and randomly cropping image patches in different locations during training. As a result, our dynamic way generates more training data than fixed way, which is crucial for deep model to avoid overfitting. We conduct an experiment to compare the two different augmentation strategy. The results in Table VIII show that dynamic augmentation way gives the better result than static one (as done in [16]), as expected. At the same time, the result is worst without augmentation obviously. Please note that here without augmentation

means without special augmentation strategy, we perform data augmentation only by cropping nine patches from each image and flipping them. We simply fix the nine cropped points as top, center, and bottom combining with left, center, and right. Each patch is 25% of the original size.

V. Conclusion

In this paper, we proposed a deep CNN for crowd counting. We introduced a dynamic data augmentation strategy to train deep CNN with existing limited crowd counting dataset. In order to decrease the overfitting caused by limited training samples, multitask learning was employed to combine different dataset with significant commonalities and learn generalizable representations across them. We also empirically found that aggregating information about the statistics of local descriptors aggregated over the image can be beneficial for this task. To this end, NetVLAD was introduced in our system which can be jointly optimized via efficient backpropagation. In order to address the scale variation problem, we proposed to use multiscale features from both bottom and top layers within the "deeply supervised" strategy. In this way, multiple features with different receptive of fields, which correspond to object with different scales, can automatically learn to contribute in the final counting results. Much improved results on three different challenging datasets with different crowd density and scenes established the superiority of the proposed method over existing state of art.

References

[1] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Crowd counting using multiple local features," in *Proc. Digit. Image Comput., Techn. Appl.*, 2009, pp. 81–88.

[2] M. S. Parwez, D. B. Rawat, and M. Garuba, "Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2058–2065, Aug. 2017.

[3] K. Chen, S. Gong, T. Xiang, and C. Change Loy, "Cumulative attribute space for age and crowd density estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2467–2474.

[4] D. Ryan, S. Denman, S. Sridharan, and C. Fookes, "An evaluation of crowd counting methods, features and regression models," *Comput. Vision Image Understanding*, vol. 130, pp. 1–17, 2015.

[5] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert, "Density-aware person detection and tracking in crowds," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2423–2430.

[6] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 3–10.

[7] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 833–841.

[8] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 589–597.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[10] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[11] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1–9.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Int. Conf. Learn. Representations*, arXiv:1409.1556, 2015.

[13] Z. Shi, Y. Ye, and Y. Wu, "Rank-based pooling for deep convolutional neural networks," *Neural Netw.*, vol. 83, pp. 21–31, 2016.

[14] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 1299–1302.

[15] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor, "Fully convolutional crowd counting on highly congested scenes," *Int. Conf. Comput. Vision Theory Appl.*, arXiv: 1612.00220, 2017.

[16] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in *Proc. ACM Int. Conf. Multimedia*, 2016, pp. 640–644.

[17] D. Onoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 615–629.

[18] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, "Multi-scale convolutional neural networks for crowd counting," *IEEE Int. Conf. Image Process.*, arXiv:1702.02359, 2017.

[19] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 160–167.

[20] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3119–3127.

[21] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 2809–2813.

[22] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5297–5307.

[23] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2547–2554.

[24] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. 18th Int. Conf. Artif. Intell. Statist.*, 2015, pp. 562–570.

[25] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[26] B. Julesz, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, no. 5802, pp. 91–97, 1981.

[27] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–7.

[28] C. C. Loy, K. Chen, S. Gong, and T. Xiang, "Crowd counting and profiling: Methodology and evaluation," in *Modeling, Simulation and Visual Analysis of Crowds*. New York, NY, USA: Springer, 2013, pp. 347–382.

[29] S. A. M. Saleh, S. A. Suandi, and H. Ibrahim, "Recent survey on crowd density estimation and counting for visual surveillance," *Eng. Appl. Artif. Intell.*, vol. 41, pp. 103–114, 2015.

[30] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3304–3311.

[31] C. E. Rasmussen, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[32] L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, and M. Pietikainen, "A survey of recent advances in texture representation," *CoRR*, abs/1801.10324, 2018.

[33] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[34] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3384–3391.

[35] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1–8.

[36] R. Arandjelovic and A. Zisserman, "All about VLAD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1578–1585.

[37] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Representations*, 2016.

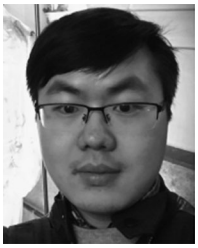
[38] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 241–257.

[39] J. Wang, Y. Sun, W. Zhang, I. Thomas, S. Duan, and Y. Shi, "Large-scale online multitask learning and decision making for flexible manufacturing," *IEEE Trans. Ind. Informat.*, vol. 12, no. 6, pp. 2139–2147, Dec. 2016.

[40] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1324–1332.

[41] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 689–692.

[42] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
[43] J. Pan, E. Sayrol, X. Giro-i Nieto, K. McGuinness, and N. E. O'Connor, "Shallow and deep convolutional networks for saliency prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 598–606.
[44] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. Freeman, "Sift flow: Dense correspondence across different scenes," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 28–42.
[45] S.-F. Lin, J.-Y. Chen, and H.-X. Chao, "Estimation of number of people in crowded scenes using perspective transformation," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 31, no. 6, pp. 645–654, Nov. 2001.



Zenglin Shi received the Bachelor's degree engineering in computer science and Master's degree engineering in computer science from the School of Zhengzhou University, Zhengzhou, China, in 2014 and 2017, respectively. He is currently working toward the Ph.D. degree at the University of Amsterdam, Amsterdam, The Netherlands.

His research interests include machine learning, computer vision, and deep learning.

Le Zhang received the B.Eng. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2011 and the M.Sc. degree from Nanyang Technological University, Singapore, in 2012. He was with Nanyang Technological University for his Ph.D. degree.

He is now with the Advanced Digital Science Center, Singapore. His research interest includes pattern classification and computer vision.

Yibo Sun received the Bachelor's degree and the Master's degree from the School of Zhengzhou University, Zhengzhou, China, in 2015 and 2018, respectively.

His research interests include pattern recognition and deep learning.

Yangdong Ye received the Ph.D. degree from China Academy of Railway Sciences, Beijing, China.

He is currently a Professor with Zhengzhou University, School of Information Engineering. He worked one year as a senior visiting scholar in Deakin University, Australia. He has wide research interests, mainly including machine learning, pattern recognition, knowledge engineering and intelligent system. He has published some papers in peer-reviewed prestigious journals and conference proceedings, such as IEEE TRANSACTIONS ON MULTIMEDIA, IEEE MULTIMEDIA, NEURAL NETWORKS, IEEE CVPR, IJCAI, and ACM Multimedia. More details about his research and background can be found at http://www5.zzu.edu.cn/mlis/.