Global Context Instructive Network for Extreme Crowd Counting

YUANYUAN WANG, XING WANG, XIANGZHI BAI, TAO ZHAO, AND YONGCE CHENG

1Image Processing Center, Beijing University of Aeronautics and Astronautics, Beijing 100191, China
2Science and Technology on Complex System Control and Intelligent Agent Cooperation Laboratory, Beijing 100074, China
3State Key Laboratory of Virtual Reality Technology and System, Beijing University of Aeronautics and Astronautics, Beijing 100191, China
4Advanced Innovation Center for Biomedical Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China
5The 3rd Research Institute of China Electronics Technology Group Corporation, Beijing 100015, China

Corresponding authors: Xing Wang (wangxingtju@126.com) and Xiangzhi Bai (jackybxz@buaa.edu.cn)

This work was supported by the National Key Research and Development Program of China (2019YFB1311301).

ABSTRACT
Crowd counting has gained popularity due to wide applications, such as intelligent security, and urban planning. However, scale variation and perspective distortion make it a challenging task. Most existing works focus on multi-scale feature extraction to address the challenge of scale variation and perspective distortion. In this paper, we propose a novel Global Context Instructive Network (GCINet), which devotes to making full use of extracted features and obtaining precise counts. The main contributions are four folds. First, we construct a three-column Feature Processor to generate features with different scales. Second, an Instructive Module is proposed to introduce global context which is the substance for generating adaptive features. Based on global context, the three-column Feature Processor constitutes an adaptive feature generator. Third, a novel loss function which integrates Euclidean distance and spatial correlation is proposed to enhance the spatial correlation and consistency between pixels. We no longer regard a pixel as an independent point in the calculation, but consider the neighborhood space of the pixel to achieve complementary effects. Finally, we conduct experiments on ShanghaiTech dataset, UCF_CC_50 dataset, UCF_QNRF dataset and UCSD dataset which show that our approach achieves state-of-the-art performance.

INDEX TERMS
Crowd counting, density map, adaptive generator, global context instructive network (GCINet).

I. INTRODUCTION
With the development of urbanization, human social activities are increasingly frequent. This increases the probability of congestion. When the crowd density exceeds the safety limit, the chance of a stampede event increases greatly. In addition, it can cause other potential hazards, such as traffic congestion, which gives a great threat to social security. Crowd counting becomes even more important. However, massive challenges exist in crowd analysis, such as occlusion, high aggregation, uneven distribution of people, complex background, scale variation and perspective distortion.

The study of crowd counting and density estimation play critical roles in the maintenance of social security. And relevant advanced methods have been proposed by researchers. Early crowd counting methods are mainly based on detection [1]–[3] and regression [4]–[6]. However, both kinds of the methods mentioned above failed in dense conditions. Recently, methods based on density estimation [7]–[10] have been a new research hotspot among researchers. By deploying density estimation-based method, we can obtain the location of each people by pixel-wise regression and obtain the crowd counts through integral.

Hand-crafted representations were often used in early density estimation-based methods, whereas these methods usually failed in extremely dense crowds. Attracted by the strong ability of feature extraction, researchers have introduced Convolutional Neural Networks (CNN) into density estimation method for crowd counting [10]–[16]. The effectiveness of density estimation has been proved by CNN representations. Although the effectiveness of CNN-based density map estimation is superior to the effectiveness of traditional methods, it turns out that most methods predict less than real counts in extremely dense scenes with large scale
variation. To overcome the problem shown in Fig. 1, various approaches [16]–[18] have been proposed. These methods have made improvements by deploying multi-column network structures. The features learned by multi-column CNN are adapted to human head of different sizes in the same scene. This architecture has achieved better performance compared with single column. This shows multi-scale features play an important role in crowd counting.

However, there exist two issues in CNN density estimation-based crowd counting methods. Most works try to address scale variation problem and focus on extracting multi-scale features. The first issue is that they ignore decoding high-quality density maps from multi-scale features adaptively. The other problem is that most methods treat pixels as isolate by deploying Euclidean distance as loss function. Euclidean distance plays an important role in crowd counting by measuring the difference between output density map and ground truth at the pixel-level. However, it assumes pixels are independent from each other [20]. For example, head is an area consisting of several pixels instead of a single pixel. Thus, the neighborhood of a pixel should be considered when determining the position of the head.

In order to mitigate the above problems existing in the current crowd counting network and obtain better performance, an instructive network named Global Context Instructive Network (GCINet) is proposed in this paper. The overall design architecture is shown in Fig. 2. Different from previous works, we make an effort to design an instructive network for density map decoder, and we not only focus on mining multi-scale features, but also dedicate to finding out how to obtain more accurate density maps from extracted features, that is, to make the features adaptive.

In the initial feature extraction, we deploy the first 13 layers of pre-trained VGG16 [19] as the feature extractor. There are two models commonly used to extract features, VGG16 and ResNet [20]. We verify both by experiments and find that backbone has an important impact on the experimental results. Finally, we select VGG as backbone for initial feature extraction. Density map decoder consists of three parts: three-column Feature Processor, Instructive Module, and Merging Module. three-column Feature Processor is composed of three parallel columns with different filter sizes. Instructive Module is designed to introduce global context guidance information. For the first issue, we deploy three-column Feature Processor to generate high-level features with the instruction of global context, then we use Merging Module to get the final density map. For the second issue, we propose a novel distance measurement, which integrates Euclidean distance and spatial correlation loss. Euclidean distance measures the difference between the output and...
ground truth pixel-wise, while spatial correlation loss takes advantage of spatial information.

Our model takes the whole image as an input, and generates high-level features under the guidance of global context information, and finally generates crowd density map. The final density map has a resolution of one-sixteenth of the original image. Experiments have shown that our method is superior to existing methods in counting accuracy.

This paper mainly makes following contributions.

1) We propose a three-column Features Processor which is composed of three columns convolutional neural network. It allows the network to extract multi-scale features to the same image.

2) We establish an additional column named Instructive Module to implement low-level features for capturing global context. With the instruction of global context information, the network generates adaptive features.

3) We propose a novel distance measurement, which integrates Euclidean distance and spatial correlation loss (SCL). The former guarantees pixel-wise accuracy, while the latter preserves spatial correlation.

4) We conduct experiments to verify the performance of the proposed network. Our model has been proved to be superior to state-of-art methods on ShanghaiTech dataset, UCF_CC_50 dataset, UCF_QNRF dataset and UCSD dataset.

II. RELATED WORK
A variety of methods have been proposed to solve crowd counting problems in images and videos. In this section, we make a brief review of these methods. These methods are roughly classified into traditional approaches and CNN-based method [21].

A. TRADITIONAL APPROACHES
Traditional crowd counting methods are broadly divided into three categories according to implement approaches [21]: (1) Detection-based approaches, (2) Regression-based approaches, and (3) Density estimation-based approaches.

In early researches, the locations and counts of crowd are mostly determined by detection-based methods which use a sliding window to detect the location of people in the scene and counts the targets together [1]–[3]. The detection-based methods have achieved a certain degree of success in the sparse case. However, due to such problems as occlusion and complex background in the dense case, these methods seem to be powerless. Part-based detection methods [22], [23] which detect part of the body are adopted to work this out. However, the effect is not very satisfactory in extreme dense crowd scene.

Regression-based methods [4]–[6] remove the computationally complex detector. However, these methods cannot obtain spatial location information. The output of these methods is simple predicted counts of crowd.

Lempitsky and Zisserman [7] have introduced a new method. This method learns a linear mapping to get the estimated density map which gives the number of each region by integral. Pham et al. [8] have introduced an approach to learn a non-linear mapping due to the intractability of learning a linear mapping.

With the efforts of the researchers, the task of crowd counting has made great progress. However, hand-crafted representations [24], [25] which are widely used in early approaches are not sufficient for the task.

B. CNN-BASED METHODS
In recent years, more works are based on CNNs, and the features extracted by these methods [19] have been verified to make great improvements in multiple fields. More researchers devote to deploying CNN-based approaches [11]–[15], [27]–[30], and varieties of methods are proposed. First, high-dimensional feature information is extracted by the powerful feature extractor. Then, the crowd density distribution map is predicted by analyzing the high-dimensional feature information. Finally, the crowd density distribution map is used to get the number of crowd.

Wang et al. [27] apply CNNs on crowd counting by adopting AlexNet network [26]. To overcome the issue that performance reduces with switching scenarios, Zhang et al. [28] proposed a network trained by two objective functions, namely crowd counting and density estimation. When applying this network to another scene, the network is fed with similar images.

In [29], Shang et al. proposed an end-to-end network which lead to complexity reduction instead of above mentioned patch-based method. Boominathan et al. [30] proposed a combination of deep and shallow fully convolutional network leading to robustness of crowd scale and perspective variation. MCNN [11] took account of three columns with receptive field at different scales. Compared with MCNN [11], Sam et al. [10] proposed a Switching-CNN which trained a Switch layer to select an optimal column for accurate density mapping.

More recently, CSRNet [12] deployed a convolutional neural network as feature extractor, then used dilated convolutional neural network as decoder to obtain density map. CAN [13] concatenated features with multiple respective field sizes and calculated the importance of each location.

Through researching on crowd counting methods, especially state-of-the-art methods, we find that most works focus on extracting multi-scale features by skip-connection [32] and multi-column structure. As discussed in Section I, multi-scale features promote the performance of network. However, the diversity of multi-scale features seems to be insufficient for existing networks in large scale variation scenes. These works are not applicable to the extracted adaptive multi-scale information. The adaptability of mentioned network is not fully utilized. Recently, some researchers [33]–[35] have tried to resolve this problem and they have begun to explore the generation of adaptive features and the effect is improved but not obvious. Besides, pixel-wise Euclidean loss is widely used in crowd counting network which regards a
pixel as isolated. We argue that the prediction of each pixel should also depend on its neighborhood.

ASD [14] contributed to adaptability by proposing an adaptive scenario discovery framework to value the weights of two different density maps generated by sparse column and dense column. PaDNet [31] committed to taking full use of extracted features by captures the global and local feature.

By researching on the existing methods, we propose Global Context Instructive Network (GCINet) which places emphasis on decoding extracted features in an adaptive way.

In terms of adaptability, we assign weights to high-level features by introducing global context. Finally, the density map is determined based on the adaptive characteristics. ASD [14] directly weights both the sparse and dense density maps and outputs the final density map. In contrast, our approach makes the network more adaptive.

In [31], the global and local features are extracted from multi-scale features. While, we operate on the initial features generated by VGG16 to obtain the global context. PaDNet is a refinement of the density map by decoding the adaptive features. Besides, we also integrate spatial correlation loss (SCL) to take into account neighborhood information.

For extreme scale variations, many methods focus on generating more multi-scale features rather than how to exploit them. There are also methods to refine the density map. To this end, we design GCINet to introduce global context, which generates adaptive features and directly outputs accurate density maps. At the same time, because the head is a region, we also introduced SCL to make use of neighborhood information.

A large number of experiments have shown that our method has achieved good results in the case of extreme scale variations, and also achieved satisfying results in sparse scenes.

### III. METHODOLOGY

In this paper, we are committed to making full use of the extracted features to obtain accurate counts. As shown in Fig. 2, our network is mainly composed of two parts: a backbone feature extractor and a density map decoder. Density map decoder consists of three modules: three-column Feature Processor, Instructive Module and Merging Module. In this section, we will introduce the proposed architecture and training details.

**A. BACKBONE**

VGG and ResNet are the widely-used network structures in feature extraction at present. The common depth of VGG is 16, while the common depth of ResNet is 18, 50 or 101. Varieties of methods based on these backbones adopt new optimization schemes and multi-model fusion, and have made improvements. The two typical models are compared and verified with a pre-trained strategy on ShanghaiTech Part A dataset. And then we determine which one is more effective for feature extraction in our method. The verification results are shown in Fig. 3.

CNN extracts rich features. Deepening the network, we can get more abstract information and lose detailed information. In fact, deeper networks do not always lead to improved performance. We need to adopt specific strategies according to different tasks.

Fig. 3 shows that, when ResNet is selected as the backbone network, as the depth of the network increases, the effect becomes better and better. At the same time, the number of the parameters increases sharply. However, the best performance of ResNet is inferior to the performance of VGG16. In the task of crowd counting, deeper network lost detail information, and VGG16 with a depth of only 16 performs better than others.

Spatial information and detail information are required in the crowd counting task. However, a deeper network leads to more semantic information and less detail information. At the same time, the residual block in ResNet makes the feature maps shuttle-shaped, that is, the number of feature maps decreases first and then increases. In the process of reducing the number of feature maps, a lot of information is lost. Therefore, VGG16 is selected as the feature extraction module in this task. RGB images as input are converted into a series of feature maps by the first 13 layers of VGG16 pretrained on ImageNet.

**B. THREE-COLUMN FEATURE PROCESSOR**

The low-level features extracted by the backbone VGG16 need to be decoded according to the characteristics of scenes.

Hence, we design a three-column convolutional neural network named three-column Feature Processor to decode low-level features according to the characteristics of different scenes. The scale variation of heads needs sufficiently various features to be represented. For dense crowds, more detailed information is needed, while for relatively sparse targets, more semantic information is needed. The three-column Feature Processor is designed to capture the both information.

As illustrated in Fig. 2, our three-column Feature Processor consists of three convolution networks. These three column convolution networks are identical in structure, but have
Based on the low-level features and we multiply high-level features. We obtain global context information. We abandon up-sampling and concatenation, directly multiply high-level features. Three-column Feature Processor considers the high-level features by global context features. Overall, we use three columns with different filter sizes to construct feature maps corresponding to the scale variation. Because a larger receptive field provides more global and advanced semantic information, while a smaller receptive field provides more local and detailed information. To get more comprehensive information, we decode features from multi-scales.

The three-column Feature Processor makes the network structure somewhat adaptive, but this adaptability is limited. It simply generates specific multi-scale features based on each image. But it does not make adaptive adjustments and changes based on the characteristics of different images. In order to make the network really adaptable, we designed an Instructive Module.

C. INSTRUCTIVE MODULE
In this subsection, we introduce global context which is utilized to guide low-level features to generate adaptive high-level features.

We regard acquiring the location of the human heads as a more elaborate semantic segmentation task. The location of each head is a class. Therefore, the density estimation-based methods also face challenge of intra-class consistency. The intra-class consistency problem is mainly caused by lack of context information. Therefore, we utilize Global Average Pooling (GAP) to introduce the global context. However, the global context only has high semantic information, which does not help to recover spatial information [36]. Therefore, we also need a multi-scale receptive field and context to refine the spatial information. This complements our three-column Feature Generator.

The global average pooling is the firstly used for the semantic segmentation task in ParseNet [37]. The work in [38] also employs this strategy and shows that the global context has a significant impact on the performance of network. Attention can help the network structure focus on what we value.

The above works use a method of up-sampling and concatenation. Our method does not adopt this strategy. We abandon up-sampling and concatenation, directly multiply the high-level features by global context features. Overall, we deploy three-column Feature Processor to obtain high-level features. Three-column Feature Processor considers more local information when decoding low-level features, but lacks of global context. We obtain global context information based on the low-level features and we multiply high-level features with global features generated by Instructive Module.

The final output is a series of scored feature maps with global context, and the calculation process can be expressed as the following equations.

\[ y_k = F(x, C_1) = \sum_{i=1}^{M} C_1 x_i \]  

(1)

In (1), \( x \) represents low-level feature maps processed by Global Average Pooling., \( k \in \{1, 2, \ldots, K\} \), where \( K \) is the number of output feature channels. \( C_1 \) is the operation of convolutional network with \( 1 \times 1 \) kernel size. \( M \) is the number of input feature map channels, and \( x_i \) is the \( i \)-th feature maps of \( x \).

\[ \delta_k(y_k) = \frac{1}{1 + \exp(-y_k)} \]  

(2)

\[ Y = \delta_k x \]  

(3)

In (2), \( \delta_k \) is probability corresponding with the \( k \)-th channel. In (3), \( Y \) represents the final output.

These three formulas represent Global Average Pooling followed by convolution layers, activation function Sigmoid and multiplication respectively.

D. MERGING MODULE
Under the guidance of the Instructive Module, the low-level features generated by VGG16 are calculated by three-column Features Processor. High-level features with adaptability is obtained by the calculation of three-column Features Generator and Instructive Module. To get a density map, a merging module is needed to convert these adaptive feature maps into a density map. Therefore, we design a feature merging module to generate a crowd density map by calculating high-level features.

We construct a merging module consisting of two convolutional layers with a kernel sizes of \( 1 \times 1 \) and \( 3 \times 3 \). the kernel size of the first convolutional layer is used to reduce the number of channels. Finally, the feature maps are calculated by the final convolutional layer which outputs a crowd density map. in most works [14], [45], the final output is obtained by simple \( 1 \times 1 \) convolution layer, and this operation is functionally equivalent to a fully connected layer. It only considers the value of a single pixel, while the \( 3 \times 3 \) convolution kernel operates on the neighborhood and contains more spatial information.

E. LOSS FUNCTION
The number of crowd is calculated by integrating all the regions of the density map. To get accurate counts, Euclidean distance is usually used to limit the distance between the output of density map and the ground truth. This measurement can judge the probability pixel-wise. Euclidean distance guarantees the accuracy of results on pixel level. However, this measurement assumes that the pixels are isolated and ignore neighborhood relationships. For the location of the
head in the image, the head occupies an area instead of a pixel. Therefore, we propose a novel distance measurement which integrates spatial correlation loss (SCL) and Euclidean distance to achieve complementary effects. Our loss function is described as follow:

\[ L = L(\theta) + \lambda L_{SCL} \]  

(4)

where \( \lambda \) is a parameter of balancing the weight of two functions. \( L(\theta) \) is the Euclidean distance between predicted density map and ground truth. \( L_{SCL} \) proposed in [33] is the spatial correlation loss.

\[ L(\theta) = \frac{1}{2N} \sum_{i=1}^{N} \|D(X_i; \theta) - D(X_i)\|^2 \]  

(5)

where, \( D(X_i; \theta) \) is the predicted density map of GCINet., \( \theta \) is the parameters of the model. \( D(X_i) \) is the ground truth density map of image \( X_i \), and \( N \) is the batch size of training.

\[ L_{SCL} = \frac{1}{N} \sum_{i=1}^{N} \left[ 1 - \frac{\sum_{p \in Q} \sum_{q \in Q} (O_{pq} \cdot G_{pq})}{\sqrt{\sum_{p \in Q} \sum_{q \in Q} O_{pq}^2 \cdot \sum_{p \in Q} \sum_{q \in Q} G_{pq}^2}} \right] \]  

(6)

where \( O_{pq}, G_{pq} \) represent pixels of output density map and ground truth density map, respectively. Each density map has \( P \times Q \) pixels.

F. TRAINING DETAILS

1) GROUND TRUTH GENERATION

Since the head position is an area and the calibration data is absolute coordinate, the direct use of calibration data will cause inaccurate results. Therefore, density maps that can be applied to crowd counting tasks need to be transformed according to the marked coordinates. The density map generation method [18] used in this paper is described in detail below. Density map is defined by convolution of impulse function with Gaussian kernel. Assuming the position of the annotation point is \( x \), the label with \( N \) head can be expressed as:

\[ H(x) = \sum_{i=1}^{N} \delta(x - x_i) \]  

(7)

\[ D(x) = H(x) \otimes G_{\sigma_i}(x) \]  

(8)

where, \( x_i \) indicates the pixel location of the human head. \( \delta(x - x_i) \) represents the impulse function of the head position in the image. \( N \) is the total number of heads in an image. \( G_{\sigma_i}(\bullet) \) is a Gaussian function whose standard deviation is \( \sigma_i \).

2) EVALUATION METRICS

In the task of crowd counting, we use Mean Absolute Error (MAE) and Mean Squared Error (MSE) for evaluation which are defined as follows:

\[ MAE = \frac{1}{N} \sum_{i=1}^{N} \left| C_i - C_i^{GT} \right| \]  

(9)

\[ MSE = \frac{1}{N} \sum_{i=1}^{N} \left( C_i - C_i^{GT} \right)^2 \]  

(10)

where \( N \) is the number of testing images. \( C_i \) and \( C_i^{GT} \) represent estimated count, ground truth count of the \( i \)-th image, respectively. \( C_i \) is the integral of the output density map.

3) DATA PREPROCESSING

In data preprocessing, we crop the image into 400 × 400 to achieve data augmentation. For some images whose sizes are less than 400 in length and width, such as ShanghaiTech dataset [18], it is firstly resized to 420 × 420, and then cropped. In each epoch, we randomly select a point as the top left corner of the image to be cropped. After cropping, the image is randomly flipped horizontally, and finally random noise is added. For the UCSD dataset, it provides region of interest (ROI). We first cover the image with ROI. Since the resolution of the image is too small, we then resize it to four times of its original resolution. Finally, we randomly crop a 400 × 400 patch from the images.

IV. EXPERIMENT

In this section, we mainly introduce the experimental process and results. Our experiments mainly include following parts: efficiency of Instructive Module, parameter selection of loss function and evaluation on four challenging datasets: ShanghaiTech dataset [11], UCF_CC_50 dataset [39], UCF_QNRF dataset [9] and UCSD dataset [5]. The overall results are shown in Table 1 and Table 2. In addition, the dataset used in the comparative experiments are all ShanghaiTech dataset Part A.

In the training process, Adam optimizer with learning rate of 1e-4 is adopt to optimize parameters of our model. Batch size is set to 16.

A. DATA DESCRIPTION

1) SHANGHAITECH DATASET

Kang and Chan [18] introduced a large crowd counting dataset. The dataset contains a total of 1,198 images and 330,165 head annotations. The dataset is divided into two parts: Part A and Part B. The 482 images of Part A are randomly selected from the Internet. Part B is obtained from dense Shanghai street. In contrast, Part A image scenes vary widely and are dense, while images in Part B are sparser.

2) UCF_CC_50 DATASET

This is the first challenging dataset created from a public website with different density levels, scenarios, and distorted perspectives. The authors collected images of different scenes, including concerts, protests, stadiums and marathons to maximize the diversity of scene types. The dataset contains a
TABLE 1. Estimation errors on shanghaitech dataset and UCF_CC_50 and UCF_QNRF.

| Method      | Part A | Part B | UCF_CC_50 | UCF_QNRF |
|-------------|--------|--------|-----------|----------|
|             | MAE    | MSE    | MAE       | MSE      | MAE      | MSE      |
| MCNN [11]   | 110.2  | 173.2  | 26.4      | 41.3     | 377.6    | 509.1    | 277      | 426      |
| Switching CNN [10] | 90.4  | 135.0  | 21.6      | 33.4     | 318.1    | 439.2    | 228      | 445      |
| CSRNet [12] | 68.2   | 115    | 10.6      | 16.0     | 266.1    | 397.5    | -        | -        |
| PACNN [39]  | 66.3   | 106.4  | 8.9       | 13.5     | 267.9    | 357.8    | -        | -        |
| ASD [14]    | 65.6   | 98.0   | 8.5       | 13.7     | 196.2    | 270.9    | -        | -        |
| TEDnet [40] | 64.2   | 109.1  | 8.2       | 12.8     | 249.4    | 354.5    | 113      | 188      |
| Idrees et al. [41] | -     | -      | -         | -        | -        | -        | 132      | 191      |
| ADCrowdNet [42] | 63.2  | 98.9   | 7.6       | 12.9     | 257.1    | 363.5    | -        | -        |
| CAN [13]    | 62.3   | 100.0  | 7.8       | 12.2     | 212.2    | 243.7    | 107      | 183      |
| SFCLN [44]  | 64.8   | 107.5  | 7.6       | 13.0     | 214.2    | 318.2    | 102      | 171      |
| SPN [25]    | 61.7   | 99.5   | 9.4       | 14.4     | 259.2    | 335.9    | -        | -        |
| GCINet (Ours) | 57.9  | 92.2   | 6.9       | 11.5     | 179.8    | 262.4    | 104      | 169      |

The symbol ‘-’ represents that the quoted article does not experiment on the corresponding dataset.

TABLE 2. Estimation errors on UCSD dataset.

| Methods       | MAE | MSE |
|---------------|-----|-----|
| Switching CNN [10] | 1.62 | 2.10 |
| Zhang et al. [2] | 1.60 | 3.31 |
| CSRNet [12]    | 1.16 | 1.47 |
| GCINet (Ours)  | 1.14 | 1.43 |
| MCNN [11]      | 1.07 | 1.35 |
| SPN [25]       | 1.03 | 1.32 |

total of 50 images with different resolutions and marks a total of 63,075 people. Each image has an average of 1,280 heads. However, the number of people in the image ranges from 94 to 4,543, and the density of the images varies significantly. This dataset has one major drawback: only a limited number of images are available for training and evaluation. Considering the small number of dataset images, we test using a five-fold cross-validation method.

3) UCF_QNRF DATASET

UCF_QNRF is the largest dataset used to train and evaluate crowd counting in terms of the number of annotations. It contains a total of 1,535 images. The training set contains 1201 images, and the test set contains 334 images. Because there are more orders of magnitude annotated people in crowd scenes, it is better suited to training very deep convolutional neural networks (CNN). UCF_QNRF is a new dataset, only a few methods are validated on the dataset.

4) UCSD DATASET

The UCSD dataset contains 2000 frames taken from the video and each frame has a size of 238 × 158. The dataset has a relatively low density population, averaging about 15 people in a frame. In addition, the dataset also provides region of interest (ROI). In the experiment, we divided it into a training set and a test set. The training set contains frames with an index of 600-1399 and the remained 1200 images as a test set. Since the dataset was taken from a fixed location, there is no change in the scene perspective in the image.

B. EFFICIENCY OF INSTRUCTIVE MODULE AND MERGING MODULE

In the Section III, we briefly described the Instructive Module and described the construction principles and meanings in detail. However, no actual verification has been carried out. In this section we will briefly describe the experimental process and conduct experiments to verify. The first group of experiment uses the network with the Institutional Module, while the other group removes Institutional Module. The results of two models with same parameter $\lambda = 1$ are shown in Fig. 4 (a). It is shown that three-column Feature Processor with Instructive Module improves the counting accuracy by 2.3(3.97%) in terms of MAE metric comparing with three-column Feature Processor, and enhance the counting accuracy by 8.6 (9.32%) in terms of MSE metric.

It turns out that Instructive Module improves the performance of the entire network. Instructive Module mainly obtains global context information from low-level features. Our network uses global context information as an indicator to guide the generation of high-level features.
For Merging Module, the convolution layer with only $1 \times 1$ kernel size is widely used to merge features and obtain density map. However, we deploy the convolution layer with $1 \times 1$ and $3 \times 3$ kernel size in our model. Neighborhood should also be considered, even for the last module of network. Effectiveness of convolution with a $3 \times 3$ kernel is demonstrated by experiment. As shown in Fig. 4(b). The results in this group show that Merging Module used convolution with $1 \times 1$ and $3 \times 3$ kernel sizes improves the counting accuracy by 2.8 (4.84%) in terms of MAE metric comparing with Merging Module used convolution with only $1 \times 1$ kernel size, and enhance the counting accuracy by 7.9 (8.5%) in terms of MSE metric.

**C. LOSS FUNCTION ABLATION AND PARAMETER SELECTION**

As shown in Fig. 4(d) and Fig. 4(e), when we perform experiment on the ShanghaiTech Part A, the performance of TEDNet [40] and our model have a similar trend according to different values of parameter $\lambda$. MAE metrics decreases first and then increase with $\lambda$ increasing. But there is still a slight difference. MAE metrics decreases first and then increase with $\lambda$ increasing. When $\lambda = 0.1$, our model performs the best in MAE. Factor $\lambda = 1$ is the optimal value for MSE. And when $\lambda = 1$, the MAE is just 0.4% (0.24) higher than that when $\lambda = 0.1$. Hence, $\lambda = 1$ is adopted in our model.

It is worth noting that, when we perform experiment on the ShanghaiTech Part A, in terms of the MAE metrics, if we deploy only Euclidean distance as loss function, our model still performs better than other state-of-the-art methods. The evaluation error is shown in Fig. 4(c). It is demonstrated that our model is effective in decoding low-level features. The performance of our model in MSE is shown in Fig. 4(f).

In particular, when $\lambda = 0.1$ or $\lambda = 1$, the network performs better than that $\lambda = 0$, which suggests that the introduction of the SCL loss function has a significant impact on network performance.

**D. EVALUATION AND COMPRISON**

Proved by the above experiment, we use the network with Instructive Module and choose parameter $\lambda = 1$ to verify performance on three datasets: ShanghaiTech dataset, UCF_CC_50 dataset, UCF_QNRF dataset and UCSD dataset. It is demonstrated that our method performs well and exceeds state-of-the-art methods.

The performance of our method on the ShanghaiTech dataset is shown in Table 1. We achieve an improvement of 10.3 (15.1%) compared to CSRNet [12] and 3.8 (6.2%) compared to state-of-the-art SPN [25] in MAE. Fig. 5 shows the density maps generated by our model. The left images belong to Part A and the right one belongs to Part B. It is demonstrated that our model effectively responds to scale changes, and also has a good effect on the complex background scene. Our model adapts to dense scenes and sparse scenes at the same time.

In Fig. 6, we display the predicted value and the ground truth of each test image on two models (CSRNet and our model) for ShanghaiTech dataset Part A. In Fig. 7, we display the predicted value and the ground truth of each test image on two models for ShanghaiTech dataset Part B. The reasons we choose ShanghaiTech dataset are diversity of scene types and different density levels. This dataset is representative. There are two reasons for choosing CSRNet. First, the structures are similar, and the features extracted by VGG are decoded to obtain accurate results. Second, the performance of CSRNet is relatively good and has a certain representativeness.
The counts increase from left to right according to the ground truth. It can be seen intuitively that the predicted values of our model are usually accurate and the absolute deviation of our model is usually smaller. Our model is adaptable to different density levels and scenarios.

However, due to the small number of high-density image samples in the training set, when the ground truth of image is greater than 1000, our model estimation error becomes larger, and the prediction result is often smaller than ground truth. In [42], when the ground truth of image is greater than 600, their estimation error becomes larger. In contrast, our model has a good performance improvement in extremely dense situations. In contrast, our model has a good performance improvement in extremely dense situations. Our model has achieved the best results in both accuracy and adaptability.

For UCF_CC_50, the MAE and MSE value of the proposed method comparing with other state-of-the-art methods are shown in Table 1. Our approach has achieved the best performance in MAE. Comparing various methods, they usually cannot retain the same performance through different datasets. This is mainly due to different characteristics of the dataset, which requires the network having better adaptability. We select two typical networks (CSRNet and SPN) in Table 1 for analysis, it can be seen that these two networks perform well on the ShanghaiTech dataset, but in contrast, UCF is inferior compared with them. Focused on decoding similarly, our model is more adaptive under the guidance of global context information.

UCF_QNRF is a new dataset, only a few methods are validated on the dataset. We compare with resent methods on UCF_QNRF. Table 1 shows the evaluation errors for our model and other models. Our network beats the second best approach in MSE, and shows comparative performance in MAE.

Table 2 shows the effect of our network on UCSD dataset. As can be seen from the table, our experimental results are superior to other methods except MCNN in MAE and MSE. This shows that our method can not only be used in extremely dense scenes, but also achieves good results in sparse scenes.
V. CONCLUSION

In this paper, we have presented an end-to-end training network called GCINet for crowd counting and density map generation. The architecture of GCINet consists of a low-level extractor and a density map decoder. We use Adaptive Features Processor to decode the low-level features generated by VGG16. Then, we design the Instruction Module to get global context information from low-level features. We verify the effectiveness of Instruction Module, Merging Model, and proposed loss function on ShanghaiTech dataset Part A. Finally, we construct experiments on ShanghaiTech dataset, UCF_CC_50 dataset, UCF_QNRF dataset and UCSD dataset to show that our model obtains state-of-the-art performance.

REFERENCES

[1] B. Leibe, E. Seemann, and B. Schiele, “Pedestrian detection in crowded scenes,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), San Diego, CA, USA, Jul. 2005, pp. 878–885.
[2] M. Li, Z. Zhang, K. Huang, and T. Tan, “Estimating the number of people in crowded scenes by MID based foreground segmentation and head-shoulder detection,” in Proc. 19th Int. Conf. Pattern Recognit., Tampa, FL, USA, Dec. 2008, pp. 1–4.
[3] W. Ge and R. T. Collins, “Marked point processes for crowd counting,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Tampa, FL, USA, Jun. 2009, pp. 2913–2920.
[4] N. Paragios and V. Ramesh, “A MRF-based approach for real-time subway monitoring,” in Proc. IEEE Conf. Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Dec. 2001, pp. 1034–1040.
[5] A. B. Chan, Z.-S. John Liang, and N. Vasconcelos, “Privacy preserving crowd monitoring: Counting people without people models or tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Anchorage, AK, USA, Jun. 2008, pp. 1–7.
[6] Z. Cheng, L. Qin, Q. Huang, S. Yan, and Q. Tian, “Recognizing human group action by layered model with multiple cues,” Neurocomputing, vol. 136, pp. 124–135, Jul. 2014.
[7] V. Lempitsky and A. Zisserman, “Learning to count objects in images,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Vancouver, BC, Canada, Dec. 2010, pp. 1324–1332.
[8] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, “COUNT forest: CO-voting uncertain number of targets using random forest for crowd density estimation,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Boston, MA, USA, Dec. 2015, pp. 3253–3261.
[9] 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 Proc. Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, Sep. 2018, pp. 532–546.
[10] D. B. Sam, S. Surya, and R. V. Babu, “Switching convolutional neural network for crowd counting,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 5744–5752.
[11] 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. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 589–597.
[12] Y. Li, X. Zhang, and D. Chen, “CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, Utah, USA, Jun. 2018, pp. 1091–1100.
[13] W. Liu, M. Salzmann, and P. Fua, “Context-aware crowd counting,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Los Angeles, CA, USA, Jun. 2019, pp. 5099–5108.
[14] X. Wu, Y. Zheng, H. Ye, W. Hu, J. Yang, and L. He, “Adaptive scenario discovery for crowd counting,” in Proc. IEEE Conf. On Acoust., Speech Signal Process. (ICASSP), Brighton, U.K., May 2019, pp. 2382–2386.
[15] M. Hossain, M. Hosseinzadeh, O. Chanda, and Y. Wang, “Crowd counting using scale-aware attention networks,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Honolulu, HI, USA, Jan. 2019, pp. 1280–1288.
[16] V. A. Sindagi and V. M. Patel, “Generating high-quality crowd density maps using contextual pyramid CNNs,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy, Oct. 2017, pp. 1879–1888.
[17] X. Bai, S. Gao, and X. Cao, “Crowd counting using scale-aware attention networks,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Honolulu, HI, USA, Jan. 2019, pp. 1280–1288.
[18] D. Kang and A. B. Chan, “Crowd counting by adaptively fusing predictions from an image pyramid,” in Proc. Brit. Mach. Vis Conf. (BMVC), Newcastle, U.K., Sep. 2018, p. 89.
Y. Wang et al.: Global Context Instructive Network for Extreme Crowd Counting

[19] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[21] C. C. Loy, K. Chen, S. Gong, and T. Xiang, “Scene text detection with deep convolutional networks,” in [Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Los Angeles CA, USA, Jun. 2019, pp. 3225–3234.

[22] J. Sang, W. Wu, H. Luo, H. Xiang, Q. Zhang, H. Hu, and X. Xia, “Improved crowd counting method based on scale-adaptive convolutional neural network,” IEEE Access, vol. 7, pp. 24411–24419, 2019.

[23] Q. Wang, J. Gao, W. Lin, and Y. Yuan, “Learning from synthetic data for crowd counting in the wild,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Los Angeles CA, USA, Jun. 2019, pp. 8198–8207.

[24] L. Fiaschi, R. Nair, U. Koethe and F. A. Hamprecht, “Learning to count with deep convolutional neural networks,” in Proc. Int. Conf. Pattern Recognit. (ICPR),Tsukuba, Japan, Nov. 2012, pp. 2658–2668.

[25] X. Chen, Y. Bin, N. Sang, and C. Gao, “Scale pyramid network for crowd counting,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Honolulu, HI, USA, Jan. 2019, pp. 1941–1950.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Las Vegas, NV, USA, Dec. 2012, pp. 1097–1105.

[27] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, “Deep people counting in extremely dense crowds,” in Proc. 23rd ACM Int. Conf. Multimedia, Brisbane, QLD, Australia, Oct. 2015, pp. 1299–1302.

[28] C. Zhang, H. Li, X. Wang, and Y. Yang, “Cross-scene crowd counting via deep convolutional neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Boston, MA, USA, Jun. 2015, pp. 833–841.

[29] C. Shang, H. Ai, and B. Bai, “End-to-end crowd counting via joint learning local and global count,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Phoenix, AZ, USA, Sep. 2016, pp. 1215–1219.

[30] L. Boominathan, S. S. S. Kruthiventi, and R. V. Babu, “CrowdNet: A deep convolutional network for dense crowd counting,” in Proc. ACM MultiMedia Conf., Amsterdam, The Netherlands, Oct. 2016, pp. 640–644.

[31] Y. Tian, Y. Lei, J. Zhang, and J. Z. Wang, “PaDNet: Pan-density crowd counting,” 2018, arXiv:1811.02805. [Online]. Available: http://arxiv.org/abs/1811.02805

[32] R. Bajmanyar, E. Vig, and P. Reineartz, “MRCNet: Crowd counting and density map estimation in aerial and ground imagery,” in Proc. Brit. Mach. Vis Conf. (BMVC), Wales, U.K., Sep. 2019.

[33] Y. Zhang, C. Zhou, F. Chang, and A. C. Kot, “Multi-resolution attention for efficient crowd counting,” Neurocomputing, vol. 329, pp. 144–152, Feb. 2019.

[34] Z. Zou, Y. Cheng, X. Su, J. Ji, X. Guo, and P. Zhou, “Attend to count: Crowd counting with adaptive capacity multi-scale CNNs,” Neurocomputing, vol. 367, pp. 75–83, Nov. 2019.

[35] V. A. Sindagi and V. M. Patel, “HA-CCN: Hierarchical attention-based crowd counting network,” IEEE Trans. Image Process., vol. 29, pp. 323–335, Jul. 2019.

[36] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, “Learning a discriminative feature network for semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, Utah, USA, Jun. 2018, pp. 1857–1866.

[37] W. Liu, A. Rabinovich, and A. C. Berg, “ParseNet: Looking wider to see better,” 2015, arXiv:1506.04579. [Online]. Available: http://arxiv.org/abs/1506.04579

[38] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” 2017, arXiv:1706.05587. [Online]. Available: http://arxiv.org/abs/1706.05587

[39] M. Shi, Z. Yang, C. Xu, and Q. Chen, “Revisiting perspective information for efficient crowd counting,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Los Angeles CA, USA, Jun. 2019, pp. 779–788.

[40] 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 Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Los Angeles CA, USA, Jun. 2019, pp. 6133–6142.

[41] H. Idrees, I. Saleemi, C. Seibt, and M. Shah, “Multiple-source multi-scale counting in extremely dense crowd images,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Portland, ON, USA, Jun. 2013, pp. 2547–2554.

[42] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, and H. Wu, “ADCrowd-Net: An attention-injective deformable convolutional network for crowd understanding,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Los Angeles CA, USA, Jun. 2019, pp. 3225–3234.

[43] X. Chen, Y. Bin, N. Sang, and C. Gao, “Scale pyramid network for crowd counting,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Honolulu, HI, USA, Jan. 2019, pp. 1941–1950.