SRNet: Scale-Aware Representation Learning Network for Dense Crowd Counting

LIANGJUN HUANG, LUNING ZHU, SHIHUI SHEN, QING ZHANG, AND JIANWEI ZHANG (Member, IEEE)

1School of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai 200235, China
2Department of Informatics, University of Hamburg, 20354 Hamburg, Germany

Corresponding author: Liangjun Huang (gene_huang@sit.edu.cn)

This work was supported in part by the Natural Science Foundation of Shanghai under Grant 19ZR1455300, and in part by the National Natural Science Foundation of China under Grant 61806126.

ABSTRACT Huge variations in the scales of people in images create an extremely challenging problem in the task of crowd counting. Currently, many researchers apply multi-column structures to solve the scale variation problem. However, multi-column structures usually have complex structures with large numbers of parameters and are difficult to optimize. To this end, we propose a scale-aware representation learning network (SRNet) that uses a commonly used encoder-decoder framework. An image is converted into deep features by the first ten layers of VGG16 in the encoder. Then, the features are regressed to a crowd density map via the decoder. The decoder mainly consists of two modules: the scale-aware feature learning module (SAM) and the pixel-aware upsampling module (PAM). SAM models the multi-scale features of a crowd at each level with different sizes of receptive fields, and PAM enlarges the spatial resolution and enhances the pixel-level semantic information, thereby improving the overall counting accuracy. We conduct extensive crowd counting experiments on ShanghaiTech Part_A, UCF-QNRF, and UCF_CC_50 datasets. Furthermore, to obtain the locations of each person, we conduct crowd localization experiments on UCF-QNRF and NWPU-Crowd datasets. The qualitative and quantitative results prove the effectiveness of the SRNet in dense crowd counting and crowd localization tasks.

INDEX TERMS Dense crowd counting, multi-scale feature learning, deep learning, convolution neural network.

I. INTRODUCTION

Crowd counting is a classic computer vision task that aims to automatically count the number of people from a given image. Crowd counting has a wide range of applications, such as estimating the scale and number of people at political gatherings and sports events and detecting the flow of people in stations or tourist areas. It is greatly significant in the field of public safety [1] and has drawn widespread interest from researchers.

Some early methods [2]–[4] formulated crowd counting as a pedestrian detection problem; that is, the number of people in a scene is obtained by counting the number of pedestrian detection boxes. However, in a high-density crowd scene, the crowd counting accuracy of the above methods is not satisfactory due to the occlusion of heads and shoulders. Other methods [5], [6] have been proposed to improve the accuracy by manually extracting features and using a regressor to directly regress the number of people. However, these methods ignore the location information of people. In addition, the extraction ability of the handcrafted feature is limited due to the differences in specific scenarios. Different scenarios need to be manually designed, which limits the generalization of the method. In recent years, many convolutional neural network (CNN)-based methods [7]–[15] have been proposed to solve the crowd counting problem, and these methods benefit from the powerful feature learning capabilities of deep learning. Most of the networks, such as MCNN [7], Switching-CNN [8], CAN [10], and CP-CNN [16], aim to model multi-scale variations in crowds. Some of these methods have redundant features [7], require complex structures to train multiple regressors [8], [16] or include a large number of parameters [10].

There are other network structures for solving multi-scale problems in other computer vision tasks. Image pyramids and feature pyramids are often used in object detection
tasks to extract multi-scale features. Spatial pyramid pooling methods, such as the SPPNet [17] and ASPPNet [18] or multi-scale feature fusion realized by splicing feature maps are commonly used to solve multi-scale problems in semantic segmentation tasks. However, multi-scale feature extraction usually requires a complicated model, which leads to a sharp increase in the number of parameters. To decrease the amount of calculation required while maintaining the improved accuracy, HS-ResNet [19] was recently proposed; in this model, the Hierarchical-Split Block(HSB) separation feature is divided into two groups: one is used for identity mapping, and the other is used for extracting refined features. By combining different features and enhancing the information interaction of different groups, the network offers more advantages than other complex networks in the trade-off between the parameter amount and performance.

In summary, our main contributions are as follows:

- We propose a scale-aware feature learning module with layer-level correlations that makes full use of the contextual information and multi-scale features extracted by convolutional layers with different receptive fields to perform more accurate counting regression.
- We propose a pixel-aware upsampling module to perform semantic enhancement at a fine-grained level to improve the pixel-level regression performance.
- The above two modules are used to construct the scale-aware representation learning network for dense crowd counting. Crowd counting and crowd localization experiments are carried out on mainstream datasets, and competitive results have been achieved.

The rest of this paper is organized as follows: we introduce the related work in Section II, demonstrate the details of the proposed method in Section III, describe experiments about crowd counting, crowd localization, and ablation studies about the design of SRNet in Section IV, and conclude in Section V.

II. RELATED WORK

In this section, we first briefly review the traditional crowd counting method and the CNN-based crowd counting method. Then, we summarize the common multi-scale learning methods based on deep learning networks in computer vision tasks.

A. TRADITIONAL CROWD COUNTING METHODS

1) DETECTION-BASED METHODS

Detection-based crowd counting methods primarily design a target detector to detect pedestrians in a crowd and obtain the number of pedestrians in a scene by counting the number of detection boxes. Such methods usually extract individual objects [20] or body-part features [21]–[25] to train the target detector. Since the head, shoulders, and other parts of the human body are prone to block each other, detection-based methods easily miss pedestrians. Therefore, detection-based methods are only suitable for crowd counting in sparse scenes.

2) REGRESSION-BASED METHODS

A regression-based crowd counting method is proposed to adapt crowd estimation methods to high-density scenes. The main idea is to extract crowd characteristics and train a regressor to establish a nonlinear mapping relationship between a crowd scene image and the number of people in the scene. Commonly used crowd characteristics include foreground characteristics [26] and gradient characteristics [20], and commonly used regressors include linear regression [27] and Bayes Poisson regression [28]. The crowd counting accuracy of these methods is better than that of detection-based methods, but it ignores the location distribution information of the crowd.

3) CROWD DENSITY MAP ESTIMATION METHODS

To obtain location information while calculating the number of people, Lemptisky et al. [5] first proposed a method based on density estimation. The main idea is to learn the linear mapping between local features and the corresponding density map. [29] proposes a non-linear mapping, random forest regression, by introducing a crowdedness prior and using it to train two different forests to reduce the difficulty of learning a linear mapping. However, these methods require manual extraction of features and depend on specific scenarios; thus, the ability to extract features is limited. Different scenarios need to be manually designed, which limits the generalization of the algorithm.
B. CNN-BASED CROWD COUNTING METHODS

To reduce the above problems and eliminate the disadvantages of manual feature extraction and low accuracy caused by traditional density estimation methods, an increasing number of researchers considering the powerful feature extraction function of CNNs have used CNN-based density map methods for crowd counting. Earlier heuristic models typically leverage basic CNNs to predict the density of the crowds [6], [30], [31]. Recently, many CNN-based density map methods [7], [8], [32] have been proposed that employ CNN branches with different receptive fields to extract multi-scale information. For the first time, Li et al. [9] applied dilated convolution to a crowd counting network, where the pretraining model of VGG16 [33] was used as an encoder and dilated convolution was applied to a decoder. Sindagi and Patel [34] adopted a multi-task method and designed a cascaded CNN to perform density classification and crowd counting simultaneously. Shen et al. [35] used the idea of generating a confrontation network (a generative adversarial network (GAN)) and generated a density map through the U-Net [36] structure to improve counting accuracy through the counting consistency of the subimage and the total image. Zhou et al. [37] combined a multi-scale convolutional neural network (generator) and an adversarial network (discriminator) to generate a high-quality density map and adapt crowd counting in complex crowd scenes. There are still some other methods based on CNNs: these methods include those based on attention mechanisms [38]–[49] that segment the foreground and the background to increase the expression of foreground features and suppress the expression of background features, methods based on super-resolution [50], [51] that can improve the image quality of high-density areas and thereby improve the counting accuracy, methods based on the perspective relationship of a scene [6], [12], [52]–[56] that can adjust the human scale variation by using the perspective view to eliminate the influence of the human head scale variation, and methods based on transfer learning [57]–[60] that can reduce the dependence of the network model on labeled data and improve the generalization ability of the model. In conclusion, an increasing number of methods based on CNN density estimation [16], [40], [61]–[67] have been proposed by researchers and have become mainstream crowd counting methods.

C. MULTI-SCALE FEATURE LEARNING FOR CROWD COUNTING

Scale variation is the primary problem to be solved in crowd counting; thus, the development of multi-scale feature learning methods for the density map estimation method based on a CNN has attracted the attention of researchers. Zhang et al. [7] proposed a multi-column network structure, called MCNN, that contains three branches and uses different kernel sizes (large, medium and small). Different scale features are extracted through convolution kernels of different sizes in each column to extract and fuse multi-scale features. Zeng et al. [32] proposed a MSCNN, which also uses convolution kernels of different sizes to extract features. MSCNN is different from MCNN in that its feature fusion block uses a fully convolution layer instead of simple feature stacking. Sam et al. [8] added a switching branch on the basis of MCNN, used the three branches as three CNN regressors, and then trained classifiers to select the best matching branch of the three branches to better learn different scale features. Sindagi and Patel [16] used a global context estimator to estimate the image crowd density level, a local context estimator to estimate the patch density level, and a density estimator to estimate the crowd density. The fusion-CNN at the backend of the network fuses these three estimators to predict a high-quality multi-scale crowd density map. Liu et al. [10] added a context-aware module between VGG16 and the decoder to extract multi-scale features. Chen et al. [68] proposed a scale pyramid network (SPN) that uses a shared deep single-column convolution to extract high-level multi-scale information through a scale pyramid module containing four parallel dilated convolutions with different expansion rates. In addition, many researchers have proposed various methods based on multi-scale features, such as ADCrowdNet [46], SCNet [69], RSANet [70], efficient and switchable CNN [71], and CASA-Crowd [72].

The above mentioned multi-scale methods ignore feature correlation between multiple columns as well as the problem that increasing the complexity of the network model architecture generates a large number of parameters. Different from the above methods, we propose a scale-aware representation learning network and construct a scale-aware feature learning module, which not only transfers the extracted feature information between layers and refines it but also avoids the large increase in the number of parameters caused by the increasing complexity of the network.

D. MULTI-SCALE FEATURE LEARNING FOR OTHER COMPUTER VISION TASKS

Image pyramids are often used in object detection tasks to perform convolution operations on images of different resolutions to extract multi-scale features from the same image. However, the inference cost of image pyramids is unsatisfactory. Hence, some researchers have proposed using a feature pyramid instead of an image pyramid via feature prediction (such as direct multi-scale feature prediction, or multi-scale feature fusion plus single-scale feature prediction, or multi-scale feature fusion plus multi-scale feature prediction) to realize the input of a single image and output feature maps of different resolutions.

Spatial pyramid pooling methods are commonly used to solve multi-scale problems in semantic segmentation tasks. After pooling, multi-scale features corresponding to different resolutions are generated in some methods, such as SPPNet [17] and ASPPNet [18], or multi-scale feature fusion is realized through feature map splicing in some methods, such as U-Net [36].
FIGURE 1. Overview of the methodology in this paper. 1) The input image extracts the underlying feature information through the encoder. 2) The decoder part uses SAM to extract and fuse multi-scale information and uses PAM to increase the resolution and pixel-level semantic information. 3) Finally, multi-scale features are fed into a $1 \times 1$ convolution layer to predict the crowd density map. \( \oplus \) stands for the addition operation and feature fusion. \( n \) indicates \( n \)-th convolution processing. \( \odot \) represents the concatenation operation. “split” means feature separation. \( \odot \) represents element-wise multiplication.

III. PROPOSED METHOD

Following the current mainstream methods [7]–[10], we formulate crowd counting as a pixel-level regression problem. The network uses an encoder and decoder framework. Image \( I_i \in \mathbb{R}^{3 \times H \times W} \) is input, Eq. (1) is used to generate feature maps \( F_i \in \mathbb{R}^{1 \times C \times H \times W} \), and then Eq. (2) is used to obtain a predicted density map \( G_i \in \mathbb{R}^{1 \times 1 \times H \times W} \):

\[
F_i = F_{vgg} (I_i) \in \mathbb{R}^{1 \times C \times H \times W}, \tag{1}
\]

\[
G_i = D_{decoder} (F_i) \in \mathbb{R}^{1 \times 1 \times H \times W}, \tag{2}
\]

where \( F_{vgg} \) represents the encoder and \( C = 512 \) and \( D_{decoder} \) denotes the decoder designed in this paper.

In particular, we explore a new multi-scale feature learning method called SRNet. The proposed SRNet model consists of two parts: an encoder and a decoder. The encoder part uses the first ten layers of VGG16 [33] for feature extraction. Our main work is to design the decoder part, and the details of the decoder are introduced below.

A. SCALE-AWARE FEATURE LEARNING MODULE

Multi-scale features play a crucial role in image tasks. Experiments such as [7], [8], [32] have also been used to verify their effectiveness in crowd counting tasks. The existing multi-scale methods extract information of different scales through multi-column convolution or dilated convolution. These methods do not pay attention to the connection between different scales, make each column learn features independently, and increase the number of additional parameters. In this section, we design an effective SAM. The green block in Fig. 1 shows the feature mapping process of the main part of the proposed SAM.

The main part of the module is composed of four layers, and the beginning of each layer contains a different convolution layer to extract features of different scales. Since the scale of a crowd in congested scenes changes within a relatively small range, we use convolutions with relatively small dilation rates and relatively small kernel sizes. Dilated convolution is used to increase the receptive field without sacrificing the spatial dimension of the feature map and without increasing the number of convolution layers. The input features of the module are fed into each layer after feature separation, and the number of channels becomes half of the original value. During feature extraction, the features of each layer are connected. Taking the second layer of the first SAM component as an example, the input is the feature maps \( F_0 \) with a size of \( 1 \times \frac{C}{2} \times \frac{H}{8} \times \frac{W}{8} \); this layer first performs a convolution layer (kernel = 3 and dilation = 2) and outputs feature maps \( F_2 \). Prior to this, the output feature maps \( F_1 \) of the first layer have dimensions of \( 1 \times \frac{C}{2} \times \frac{H}{8} \times \frac{W}{8} \). \( F_2 \) and the first half of \( F_1 \) are connected in the channel dimension, and the output tensor size is \( 1 \times \frac{3C}{4} \times \frac{H}{8} \times \frac{W}{8} \). Then,
the feature maps go through the convolution layer, and the
pixel enhancement (PE) submodule shown in Eq. 5. Feature
weighting can enhance the crowd feature, and the convolution
layer provides learnable parameters for weighting. The
adaptive enhancement of semantic information is achieved
through parameter adjustment and feature weighting of the
convolution layer. The size of the final output tensor is \(1 \times \frac{C}{2} \times \frac{H}{2} \times \frac{W}{2}\). Unlike the second layer, the first layer does not
need to be connected to the upper-layer feature maps, and the
output tensor dimension of the fourth layer is \(1 \times \frac{C}{4} \times \frac{H}{4} \times \frac{W}{4}\).
The equation for the entire SAM process is as follows:

\[
f_{ij} = C_{i,j}^n (h_i) \in \mathbb{R}^{1 \times C_i \times H_k \times W_k} \quad n = 1, 2, 3, 4, \quad (3)
\]

where \(i\) represents the \(i\)-th input images and \(j\) represents the
\(j\)-th layer in this module. \(h_i\) indicates the input features of
this module, and the first input feature of SAM is \(F_i\). \(C_{i,j}^n\)
indicates \(n\)-th convolution processing, in which the size of the
convolution kernel and the expansion rate are different,
and multi-scale features can be extracted, as shown in Fig. 1.

\[
1 \times C_i \times H_k \times W_k\] indicates the input feature size of
the \(k\)-th layer \((k = 1, 2, 3)\) SAM. The input feature size of SAM is
\(1 \times 256 \times \frac{H}{8} \times \frac{W}{8}\) when \(k = 1, 1 \times 128 \times \frac{H}{4} \times \frac{W}{4}\) when \(k = 2,
and \(1 \times 64 \times \frac{H}{2} \times \frac{W}{2}\) when \(k = 3\).

\[
f'_{ij} = \begin{cases} f_{ij} & \text{for } 1 \leq j \leq 1 \\ C_{i,j}^{cov} \left( \frac{1}{c} f_{ij} + \frac{1}{2} g_{ij-1} \right) & \text{for } 2 \leq j \leq 4 \end{cases} \quad (4)
\]

where \(f_{ij}\) represents the feature map after the first convolution
operation in this layer, that is, the extracted features of
different scales. \(f'_{ij}\) \(\in\mathbb{R}^{1 \times C_i \times H_k \times W_k}\) denotes the fusion of
the upper-layer feature maps and the feature map after the
first convolution operation in this layer; \(C_{i,k}\) indicates the
channel number of \(f'_{ij}\), where \(C_{i,k} = \frac{C_i}{2}\) \((j = 1, 2, 3)\) and \(C_{i,k} = \frac{C_i}{4}\) \((j = 4)\). \(g_{ij-1}\) is the output feature map of each layer.
\([\ldots, \ldots]\) indicates that the extracted features are connected in
the channel dimension, and \(C_{i,j}^{cov}\) denotes convolution
processing, which converts the number of channels. \(\frac{1}{c}\) indicates
that half of the feature map is taken according to the number of
channels.

\[
g_{ij} = C^{cov} \left( S^g \left(f'_{ij} \right) \otimes f'_{ij} + f'_{ij} \right) \in \mathbb{R}^{1 \times C_i \times H_k \times W_k}, \quad (5)
\]

where \(\otimes\) represents elementwise multiplication. \(S^g\) indicates
sigmoid. \(g_{ij}\) is the feature map after PE and represents the
output feature map of the \(j\)-th layer in SAM. \(C^{cov}\) denotes
convolution processing performed on the input feature maps,
where the size of the convolution kernel is \(1 \times 1\).

\[
g_i = \left[ \frac{1}{2} g_{i,1}, \frac{1}{2} g_{i,2}, \frac{1}{2} g_{i,3}, \frac{1}{2} g_{i,4} \right] \in \mathbb{R}^{1 \times C_k \times H_k \times W_k}, \quad (6)
\]

where \(g_i\) represents the multi-scale features fused by SAM. \(\frac{1}{2}\)
represents half of the feature map according to the number of
channels. \([\ldots, \ldots]\) denotes the concatenation operation.

**B. PIXEL-AWARE UPSAMPLING MODULE**

We use three pooling layers to extract high-level features
while reducing the number of parameters, so upsampling is
used to increase the resolution and restore the original
image size. To further refine the features of SAM, we propose
a feature upsampling module, as shown in the orange box
in Fig. 1. This module consists of a pixel attention module
and an upsampling layer. The pixel attention module uses
a \(1 \times 1\) convolution layer and a sigmoid function to add
a learnable weight coefficient to each pixel, which is then
weighted into the input features to increase pixel-level semantic
information. In addition, in the upsampling layer, we use bilinear
interpolation for upsampling to gradually increase
the resolution and restore the original size of the image.
We take \(g_i\) as the input, which is the feature map output of
SAM, and output the feature map \(u_i\). This module is defined
as \(\text{Upsampling} \rightarrow \text{Conv2d} \rightarrow \text{Pixe}
lation(Pa) \rightarrow \text{Conv2d}\), and pixel attention is expressed as:

\[
Pa (f_{in}) = S^g \left( C^{cov} (f_{in}) \right) \otimes f_{in} + f_{in} \in \mathbb{R}^{1 \times C_i \times 2H_k \times 2W_k}, \quad (7)
\]

where \(f_{in}\) represents the input feature maps and \(C^{cov}\) denotes
covolution processing on the input feature maps, where the
size of the convolution kernel is \(1 \times 1\). \(\otimes\) denotes elementwise
multiplication. \(1 \times C_k \times H_k \times W_k\) represents the input
feature size of the \(k\)-th layer \((k = 1, 2, 3)\) PAM.

**C. SCALE-AWARE REPRESENTATION LEARNING NETWORK**

We present a schematic diagram of the model in Fig. 1
and call this model SRNet, which is used for dense crowd
counting. Our model offers the following advantages. First,
SAM can effectively use the feature connections between
levels to refine and fuse the features extracted from each
layer. In particular, it does not simply connect the features
of all layer levels but splits the features and connects some
of them as feature maps. Part of the output in the module
is connected to the next layer to continue to refine the fea-
tures, thereby preventing a sudden increase in the number
of parameters caused by the increase in network layers.
Second, PAM can restore the image resolution while enhancing
pixel-level semantic information. Third, a pixel enhancement
module is also added to SAM to enhance pixel-level semantic
information and reduce the weights of background pixels.
In summary, our model not only extracts multi-scale features
while avoiding a sudden increase in the number of parameters
but also adjusts the attention of the crowd and the background
at the pixel level. Therefore, our model performs better in
the crowd counting task. Although our model can prevent
a sudden increase in parameters when the network becomes
complicated, the problem of a large number of parameters
still exists. Determining how to continue to design a lighter
model without affecting the network performance remains to
be explored in the future.
D. DENSITY MAP GENERATION AND EVALUATION METRICS

Each head positioning point in an image is processed to generate a density map by a Gaussian kernel function:

\[ F(x) = \sum_{i=1}^{M} \delta(x - x_i) \ast g_{a_i}, \quad \sigma_i = \beta \bar{d}_i, \]

where \( g_{a_i} \) represents the two-dimensional standard Gaussian kernel function, \( \delta(.) \) denotes the Dirac delta function, \( \sigma \) represents the standard deviation, and \( M \) is the total number of people \( i \), which is the sum of all pixel values in the crowd density map \( F_i \). \( \beta \) is a constant, and \( d \) represents the diameter of a human head in the image and is the same as in Zhang et al. [7]. In this paper, \( \beta = 0.3 \), and \( \bar{d}_i \) is the average distance among k neighborhoods (k = 7). The crowd counting network establishes a nonlinear transformation between the input images \( i \) and the corresponding crowd density map \( F_i \).

In crowd counting tasks, we use the mean squared error (MSE) and mean absolute error (MAE) as the metrics for comparing the performance of SRNet against other crowd counting methods. For a test sequence with \( N \) images, MSE and MAE are defined as follows:

\[
\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (\hat{G}_i - G_i)^2, \tag{9}
\]

\[
\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |\hat{G}_i - G_i|, \tag{10}
\]

E. LOCALIZATION TASK

Compared with crowd density maps, crowd localization can more accurately reflect the distribution of crowds. Therefore, we extract head positioning points from the crowd density map with 0.5 as the threshold to determine whether a point is a head position. Then, the extracted head positioning points are matched with the ground-truth points one-to-one. If the detection is within a particular distance threshold of the ground-truth location, it is recorded as a true positive (TP); if the distance is greater than the threshold, it is recorded as a false positive (FP); if there is no detection within a ground-truth location, it is treated as a false negative (FN). We use precision (Pre), recall (Rec), and F1-measure (F1) to evaluate the localization performance under the threshold.

F. LOSS FUNCTION

The L2 loss is chosen as the crowd counting task loss function:

\[
L(\lambda) = \frac{1}{2N} \sum_{i=1}^{N} \| \hat{G}(i; \lambda) - G_i \|^2_2, \tag{11}
\]

where \( \lambda \) represents the learning parameter of the crowd counting network and \( \hat{G}(i; \lambda) \) is the output of the crowd counting network. \( G_i \) is the ground truth. \( N \) represents the number of images in the training set.

The loss function of the localization task is binary cross-entropy:

\[
\text{Loss}(\hat{M}, M) = -\frac{1}{N} \sum_{i=1}^{N} (M[i] \times \log(\hat{M}[i]) + (1 - M[i]) \times \log(1 - \hat{M}[i])), \tag{12}
\]

where \( i \) denotes the \( i \)-th image of the training set. \( N \) represents the number of images in the training set. \( \hat{M} \) is the prediction result. \( M \) is the ground truth; here, we only set it to 1 or 0, representing the presence and absence of a person, respectively.

IV. EXPERIMENTS

In this section, we mainly perform density estimation tasks and crowd localization tasks. First, we describe the configuration details of our experiments. Then, we conduct the ablation study on ShanghaiTech Part_A. Afterwards, we report the results of the proposed SRNet model on three mainstream datasets. Finally, we show the results of the localization task on NWPU-Crowd and UCF-QNRF.

A. DATASETS

1) ShanghaiTech PART_A

ShanghaiTech Part_A includes 482 images from the Internet (300 images for the training and 182 images for the testing). The number of people in the images in this dataset ranges between 33 and 3139, and most of the images contain people.

2) UCF_CC_50

UCF_CC_50 [26] includes 50 images from the Internet, and there are at most 4543 people in an image. The dataset contains 63,075 labeled individuals. This dataset contains images of extremely crowded scenes. These images are mainly acquired from FLICKR. Similar to other widely accepted methods, we also use a standard 5-fold cross-validation protocol to evaluate the algorithms.

3) UCF-QNRF

UCF-QNRF [14] contains 1535 images (1201 images for training and 334 images for testing). Compared with previous crowd statistics datasets, this dataset contains more annotations in congested scenes. The crowd populations in this dataset have large variations from 49 to 12,865, which poses a challenge for crowd counting.

4) NWPU-CROWD

The NWPU-Crowd dataset was proposed in 2020 [73]. The dataset consists of 5109 images (3109 for training, 500 for validation, and 1500 for testing) and 2,133,375 annotated examples. NWPU-Crowd introduces 351 negative samples with similar texture characteristics as crowded crowd scenes, reflecting images without people, with texture characteristics similar to those of crowded scenes. The average resolution of the images in NWPU-Crowd is 2191 × 3209, which is higher than the average resolution of the images in the other datasets.
Specifically, the maximum size of an image is $4028 \times 19,044$. The crowd population in this dataset varies from 0 to 20,033, which covers a wider range of variations and thus makes the crowd counting task more challenging.

### B. IMPLEMENTATION DETAILS

Our method adopts the first ten layers of the feature layer of VGG16 [33] as feature extractors and adopts adaptive moment estimation (Adam) [74] as the optimizer. During the training process, the batch size is set to 1. The learning rate is set to 1e-6. All experiments are performed on an NVIDIA GTX 1080Ti and an Intel(R) Core(TM) i9-7900 CPU. The data preprocessing settings are based on the C-3 framework [75]. Horizontal flip is used on all images for data enhancement. The images are limited to the size of $[1024, 1024]$ on the ShanghaiTech Part_A, UCF-QNRF, and UCF-CC-50 datasets. The images are limited to the size of $[1024, 768]$ on the ShanghaiTech Part_A. During the ablation study, PAM includes both the pixel attention submodule and bilinear interpolation upsampling layer. The other parts are the same as those in SRNet. As seen from Table 1, the experimental result with PAM (MAE = 66.0 and MSE = 96.7) is better than that with bilinear interpolation upsampling (MAE = 70.1 and MSE = 106.5) due to the addition of the pixel attention submodule. The pixel-level semantic information is enhanced in the experiment with PAM, which makes the crowd counting result of the congested area more accurate. The overall performance is thereby improved.

In addition, we explore the effect of pixel attention in PAM. We replace pixel attention with channel attention and spatial attention, and other conditions remain unchanged. As shown in Table 2, the results show that pixel attention is effective at dense crowd counting tasks and surpasses the effects of the channel attention module and spatial attention module.

As shown in Table 1, the results of (1) Conv w. Up and (3) SAM w. Up in Table 1, compared to the convolution layers (kernel size = 3), SAM can truly improve the performance due to the extraction of multi-scale information. The effect of the PE submodule is also explored. As shown in Table 1, the results of (2) SAM (w/o PE) w. Up versus (3) SAM w. Up, as well as the results of (4) SAM (w/o PE) w. PAM versus (5) SAM w. PAM, indicate that the method with PE in SAM outperforms the method without PE in SAM.

### C. ABLATION STUDY ON ShanghaiTech Part_A

To validate the effectiveness of SRNet, we conduct an ablation study in ShanghaiTech Part_A. During the ablation study, we use the same encoder and different structures of decoders. The different structures of decoders are as follows: (1) Conv w. Up (2) SAM (w/o PE) w. Up (3) SAM w. Up (4) SAM (w/o PE) w. PAM (5) SAM w. PAM.

The abovementioned structures in the ablation study are described as follows: (1) Conv w. Up: PAM is reduced to convolution layers (kernel = 3), and SAM is reduced to an upsampling layer (mode = bilinear). The other parts are the same as those in SRNet. (2) SAM (w/o PE) w. Up: The PE submodule in SAM is deleted, and PAM is replaced with an upsampling layer (mode = bilinear). The other parts are the same as those in SRNet. (3) SAM w. Up: The complete SAM method is used, and PAM is replaced with an upsampling layer (mode = bilinear). The other parts are the same as those in SRNet. (4) SAM (w/o PE) w. PAM: Only the PE submodule in SAM is deleted, and the other parts are the same as those in SRNet. (5) SAM w. PAM: This is the complete SRNet model with both SAM and PAM.

#### 1) THE EFFECTIVENESS OF SAM

In this section, we analyze the effectiveness of SAM. The comparison experiments are described in detail as follows:

| Decoder          | MAE   | MSE   |
|------------------|-------|-------|
| Conv w. Up       | 432.9 | 558.9 |
| SAM (w/o PE) w. Up | 71.6  | 111.6 |
| SAM w. Up        | 70.1  | 106.5 |
| SAM (w/o PE) w. PAM | 66.6  | 100.1 |
| SAM w. PAM       | 66.0  | 96.7  |

#### 2) THE EFFECTIVENESS OF PAM

In this section, we perform comparison experiments to analyze the effectiveness of PAM. Specifically, the comparison experiment is designed as follows: all experiments adopt the first ten layers of VGG16 as the encoder. The structure of the decoder is designed with PAM and without PAM. PAM includes both the pixel attention submodule and bilinear interpolation upsampling layer, and the other experiment uses only the bilinear interpolation upsampling layer. The other parts are the same as those in SRNet. As seen from Table 1, the experimental result with PAM (MAE = 66.0 and MSE = 96.7) is better than that with bilinear interpolation upsampling (MAE = 70.1 and MSE = 106.5) due to the addition of the pixel attention submodule. The pixel-level semantic information is enhanced in the experiment with PAM, which makes the crowd counting result of the congested area more accurate. The overall performance is thereby improved.

All experiments adopt the first ten layers of VGG16 as the encoder. The structure of the decoder is designed with SAM and without SAM. In the case without SAM, the original SAM part of SRNet is replaced with the corresponding convolution layers. As shown in the results of (1) Conv w. Up and (3) SAM w. Up in Table 1, compared to the convolution layers (kernel size = 3), SAM can truly improve the performance due to the extraction of multi-scale information.

As shown in Table 2, the results of (2) SAM (w/o PE) w. Up versus (3) SAM w. Up, as well as the results of (4) SAM (w/o PE) w. PAM versus (5) SAM w. PAM, indicate that the method with PE in SAM outperforms the method without PE in SAM.

| Module                | MAE   | MSE   |
|-----------------------|-------|-------|
| Channel attention [45] | 74.9  | 109.8 |
| Spatial attention [45] | 71.7  | 116.2 |
| Pixel attention       | 66    | 96.7  |
set and two sets of SAM w. PAM components replace SAM w. PAM with convolution layers that have the same number of layers and channels. In the experiment with four groups of SAM w. PAM component sets, the PAM upsampling multiple in the fourth group of SAM w. PAM is set to 1.) Therefore, the design of SRNet is promising for dense crowd counting.

**D. RESULTS ON ShanghaiTech Part_A**

Table 4 shows the results of thirteen mainstream methods (MCNN [7], Switching-CNN [8], IG-CNN [76], CSRNet [9], TEDNet [15], DUBNet [77], PACNN [12], PSODC [78], GSP [11], MSCANet [79], PACNN+CSRNet [12], CAN [10], ASNet [38]) and SRNet (ours) on the ShanghaiTech Part_A dataset, among which eleven are pretrained methods and ten are multi-scale methods. Compared with the other pretrained methods, SRNet achieves the second-best MAE score of 66.9 and the best MSE score of 96.7. Compared with the other multi-scale methods, SRNet achieves the second-best MSE score of 96.7. Compared with the other multi-scale methods, SRNet achieves the second-best result. SRNet achieves the best MSE result when all methods use L2 loss.

**E. RESULTS ON UCF_CC_50**

Table 5 indicates the performances of the other fifteen mainstream algorithms and SRNet, among which ten are pretrained methods and thirteen are multi-scale methods. Compared with the other pretrained methods, SRNet achieves the fifth-best MAE score of 184.1 and the best MSE score.
TABLE 5. Comparison with state-of-the-art approaches on UCF_CC_50.

| Method       | Year | MAE  | MSE  | Pretrain Multi-scale |
|--------------|------|------|------|----------------------|
| MCNN [7]     | 2016 | 377.6 | 509.1 | ✓                    |
| Switching CNN [8] | 2017 | 318.1 | 439.2 | ✓                    |
| CSRNet [9]   | 2018 | 266.1 | 397.5 | ✓                    |
| PACNN [12]   | 2019 | 267.9 | 357.8 | ✓                    |
| TEDNet [15]  | 2019 | 249.4 | 354.5 | ✓                    |
| IG-CNN [76]  | 2018 | 291.4 | 349.4 | ✓                    |
| SPN [68]     | 2019 | 259.2 | 335.9 | ✓                    |
| MSCANet [79] | 2021 | 242.8 | 329.8 | ✓                    |
| PACNN+CSRNet [12] | 2019 | 241.7 | 320.7 | ✓                    |
| CAN [10]     | 2019 | 212.2 | 243.7 | ✓                    |
| AMSNet [80]  | 2020 | 208.4 | 297.3 | ✓                    |
| AMRNet [81]  | 2021 | 184   | 265.8 | ✓                    |
| LibraNet [82] | 2020 | 181.2 | 262.2 | ✓                    |
| ASNet [38]   | 2021 | 174.8 | 251.6 | ✓                    |
| M-SFANet [83] | 2021 | 167.5 | 256.3 | ✓                    |
| SRNet (ours) | 2021 | 184.1 | 232.7 | ✓                    |

TABLE 6. Comparison with state-of-the-art approaches on UCF-QNRF.

| Method       | Year | MAE  | MSE  | Pretrain Multi-scale |
|--------------|------|------|------|----------------------|
| CMTL [34]    | 2017 | 252  | 514  | ✓                    |
| Switching CNN [8] | 2017 | 228  | 445  | ✓                    |
| MCNN [7]     | 2016 | 277  | 426  | ✓                    |
| CL [14]      | 2018 | 132  | 191  | ✓                    |
| RAANet [84]  | 2019 | 111  | 190  | ✓                    |
| TEDNet [15]  | 2019 | 113  | 188  | ✓                    |
| MSCANet [79] | 2021 | 104.1| 183.8| ✓                    |
| CAN [10]     | 2019 | 107  | 183  | ✓                    |
| HACCN [13]   | 2019 | 118.1| 180.4| ✓                    |
| SRNet (ours) | 2021 | 108.2| 177.5| ✓                    |

F. RESULTS ON UCF-QNRF

Table 6 shows the performances of the proposed SRNet model and nine other algorithms. Among the ten methods, five are pretrained methods, and seven are multi-scale methods. SRNet outperforms the other pretrained methods on MSE. Compared with the other multi-scale methods, our model achieves better results (MAE of 108 and MSE of 232.7). From the above comparison, we can conclude that our SRNet model attains a competitive result (second place with respect to MAE and the best with respect to MSE) on UCF-QNRF.

G. COMPARISON OF THE PARAMETERS AND FLOPs

Current state-of-the-art network models are usually complex in design and include a large number of parameters. Our network model performs similarly to the current state-of-the-art methods with significantly fewer network parameters. We list the parameters, FLOPs and counting performance comparison on the dense crowd dataset UCF_CC_50, as shown in Table 7. The input tensor size is $1 \times 3 \times 224 \times 224$ when testing the FLOPs. The counting performance of our method is similar to that of other state-of-the-art approaches and requires the least number of parameters (9255k) and FLOPs (17.98 G), which verifies that the satisfactory performance of our model is not due to an increase in the number of parameters but the reasonable design of the network structure.

H. LOCALIZATION RESULTS ON UCF-QNRF AND NWPU-CROWD

To obtain the locations of each person, we conduct crowd localization experiments on UCF-QNRF and NWPU-Crowd. For the UCF-QNRF dataset, we calculate the values of the evaluation metrics (Pre, Rec and F1) with a radius of 0-100 pixels. Finally, we regard the average of these results as the final value [14]. For the NWPU-Crowd dataset, we follow
FIGURE 4. Visualization of the partial localization results on the NWPU-Crowd dataset. The green point is a true positive (its center is the ground-truth position, and its radius is \( r \)); the red points are false negatives inside the red circle; and the magenta points and the corresponding circles are false positives.

TABLE 8. Localization performance comparison with state-of-the-art approaches on UCF-QNRF.

| Method       | F1  | Prec | Rec  |
|--------------|-----|------|------|
| MCNN [7], [14] | 0.62 | 0.60 | 0.64 |
| ResNet [14], [85] | 0.64 | 0.62 | 0.67 |
| DenseNet59 [86] | 0.64 | 0.70 | 0.58 |
| SRNet (ours) | 0.66 | 0.63 | 0.62 |

TABLE 9. The localization performance comparison with state-of-the-art approaches on NWPU-Crowd. ‘(s)’ represents a small threshold, which means that this method applies the internally tangent circle radius of the human head ground-truth box as the threshold. ‘(l)’ represents a large threshold, which means that this method applies the circumcircle radius of the human head ground-truth box as the threshold.

| Method       | F1 (%) | Prec (%) | Rec (%) | MAE/MSR/NAE |
|--------------|--------|----------|---------|-------------|
| TinyFaces(s) [73], [87] | 59.8   | 54.3     | 66.6    | 240/4736/2.0/962 |
| TinyFaces(l) [73], [87] | 55.3   | 50.2     | 61.7    | 240/4736/2.0/962 |
| VGG+GPR(s)    | 46.0   | 49.9     | 42.7    | 105.8/504.4/0.931 |
| VGG+GPR(l)    | 56.3   | 61.0     | 52.2    | 105.8/504.4/0.931 |
| RAZNet(s) [73], [88] | 54.5   | 60.5     | 49.6    | 128/7665/4/0/508 |
| RAZNet(l) [73], [88] | 62.5   | 69.2     | 56.9    | 128/7665/4/0/508 |
| SRNet(s)      | 57.9   | 59.6     | 56.4    | 124.5/604.4/0/853 |
| SRNet(l)      | 65.3   | 67.1     | 63.5    | 124.5/604.4/0/853 |

V. CONCLUSION

In this paper, we propose a scale-aware representation learning network (SRNet) for dense crowd counting; SRNet provides a new method for extracting and fusing features of different scales. SRNet includes multiple SAM-PAM components. SAM is a scale-aware feature learning module, and PAM is a pixel-aware upsampling module. SAM extracts multi-scale features through convolutions of different scales and dilated convolutions of different scales. SAM employs the ‘split’ and ‘concat’ operations to fuse multi-scale features, which can reduce the feature redundancy. Then, the extracted features are enhanced at the pixel level. PAM uses a pixel attention module to enhance semantic information and uses bilinear interpolation upsampling to improve the resolution. A large number of experiments have proven that the proposed SRNet model achieves competitive counting performance for dense crowd scenes and offers more advantages in the trade-off between parameter amount and performance. Since SRNet can decrease the number of parameters to enhance the potential to reduce model calculation, its applications to crowd counting in video sequences can be explored in the future.

REFERENCES

[1] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L. Q. Xu, “Crowd analysis: A survey,” Mach. Vis. Appl., vol. 19, nos. 5–6, pp. 345–357, 2008.

[2] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, “Scale-aware fast R-CNN for pedestrian detection,” IEEE Trans. Multimedia, vol. 20, no. 4, pp. 985–996, Apr. 2018.
R. Stewart, M. Andriluka, and A. Y. Ng, “End-to-end people detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3127–3136.

W. Liu, S. Liao, W. Ren, W. Hu, and Y. Yu, “High-level semantic feature detection: A new perspective for pedestrian detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5187–5196.

V. Lemptsky and A. Zisserman, “Learning to count objects in images,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 1324–1332.

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. (CVPR)*, Jun. 2015, pp. 833–841.

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)*, Jun. 2019, pp. 5225–5321.

D. B. Sam, S. Surya, and R. V. Babu, “Switching convolutional network for crowd counting,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4031–4039.

Y. Li, X. Zhang, and D. Chen, “CRSNNet: Dilated convolutional neural networks for understanding the highly congested scenes,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1091–1100.

W. Liu, M. Salzmann, and P. Fua, “Context-aware crowd counting,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5099–5108.

S. Aich and I. Stavness, “Global sum pooling: A generalization trick for object counting with small datasets of large images,” 2018, arXiv:1805.11123. [Online]. Available: http://arxiv.org/abs/1805.11123

M. Shi, Z. Yang, C. Xu, and Q. Chen, “Revisiting perspective information for efficient crowd counting,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6890–6899.

V. A. Sindagi and V. M. Patel, “HA-CCN: Hierarchical attention-based crowd counting network,” *IEEE Trans. Image Process.*, vol. 29, pp. 323–335, 2020.

H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and H. Idrees, “What can help pedestrian detection?,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4706–4715.

J. Chen, W. Su, and Z. Wang, “Crowd counting with crowd attention convolutional neural network,” *Neurocomputing*, vol. 382, pp. 210–220, Mar. 2020.

Z. Long, L. Yue, J. Shen, F. Zhu, X. Zhen, X. Cao, and L. Shao, “Attentional neural fields for crowd counting,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 5714–5723.

L. Lu, H. Di, Y. Lu, L. Zhang, and S. Wang, “A two-level attention-based interaction model for multi-person activity recognition,” *Neurocomputing*, vol. 322, pp. 195–205, Dec. 2018.

Z. Shi, P. Mettes, and C. Snoek, “Counting with focus for free,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4200–4209.

Z. Duan, H. Saleem, C. Seibert, and M. Shah, “Self-supervised multi-scale learning of dense crowd images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2547–2554.

B. Liu and N. Vasconcelos, “Bayesian model adaptation for crowd counts,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4175–4183.

A. B. Chan and N. Vasconcelos, “Bayesian Poisson regression for crowd counting,” in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 545–551.

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)*, Dec. 2015, pp. 3253–3261.

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*, Oct. 2015, pp. 1299–1302.

M. Fu, P. Xu, X. Li, Q. Liu, M. Ye, and C. Zhu, “Fast crowd density estimation with convolutional neural networks,” *Eng. Appl. Artif. Intell.*, vol. 43, pp. 81–88, Feb. 2015.

L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, “Multi-scale convolutional neural networks for crowd counting,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 465–469.

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

V. A. Sindagi and V. M. Patel, “CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting,” in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surv. (AVSS)*, Aug. 2017, pp. 1–6.

Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang, “Crowd counting via adversarial cross-scale consistency pursuit,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5245–5254.

O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. Image Comput. Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

Y. Zhou, J. Yang, H. Li, T. Cao, and S.-Y. Kung, “Adversarial learning for multiscale crowd counting under complex scenes,” *IEEE Trans. Cybern.*, early access, Jan. 3, 2020, doi: 10.1109/TCYB.2019.2956091

X. Jiang, L. Zhang, M. Xu, T. Zhang, P. Lu, B. Zhou, X. Yang, and Y. Pang, “Attention scaling for crowd counting,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4706–4715.

J. Chen, W. Su, and Z. Wang, “Crowd counting with crowd attention convolutional neural network,” *Neurocomputing*, vol. 382, pp. 210–220, Mar. 2020.

Z. Duan, H. Chen, and J. Deng, “AAFM: Adaptive attention fusion mechanism for crowded scene,” *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, 2013, pp. 1341–1348.

J. Gao, Q. Wang, and Y. Yuan, “SCAR: Spatial-/channel-wise attention interaction model for multi-person activity recognition,” *Neurocomputing*, vol. 291, pp. 1–8, Oct. 2019.

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)*, Jan. 2019, pp. 1280–1288.

J. Liu, C. Gao, D. Meng, and A. G. Hauptmann, “DecisionNet: Counting varying density crowds through attention guided detection and density estimation,” in *Proc. CVPR*, Jul. 2018, pp. 5197–5206.

J. Gao, Q. Wang, and Y. Yuan, “SCAR: Spatial-Channel-wise attention regression networks for crowd counting,” *Neurocomputing*, vol. 363, pp. 1–8, Oct. 2019.

N. Liu, L. Long, C. Zou, Q. Niu, L. Pan, and H. Wu, “ADCrowdNet: An attention-injective deformable convolutional network for crowd understanding,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3229–3236.

D. Qin, K. Li, Z.-J. Zha, and M. Wang, “DADNet: Dilated-attention deformable ConvNet for crowd counting,” in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 1823–1832.

Z. Duan, H. Chen, and J. Deng, “AAMF: Adaptive attention fusion mechanism for crowd counting,” *IEEE Access*, vol. 8, pp. 138297–138306, 2020.

S. Wang, Y. Lu, T. Zhou, H. Di, L. Lu, and L. Zhang, “SCLNet: Spatial context learning network for congested crowd counting,” *Neurocomputing*, vol. 404, pp. 227–239, Sep. 2020.
Z. Yan, Y. Yuan, W. Zuo, X. Tan, Y. Wang, S. Wen, and E. Ding,

Y. Xie, Y. Lu, and S. Wang, “RSANet: Deep recurrent scale-aware network model for people counting and localization in crowd scenes,” in Proc. IEEE/ACM Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 952–961.

S. Basalamah, S. D. Khan, and H. Ullah, “Scale-driven convolutional neural network model for people counting and localization in crowd scenes,” IEEE Access, vol. 7, pp. 71576–71584, 2019.

S. D. Khan, H. Ullah, M. Uzair, M. Ullah, R. Ullah, and F. A. Cheikh, “Disam: Density independent and scale aware model for crowd counting and localization,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2019, pp. 4474–4478.

J. Gao, Q. Wang, and X. Li, “PCC Net: Perspective crowd counting via spatial convolutional network,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 10, pp. 3486–3498, Oct. 2020.

A. B. Chan, Z.-S. John Liang, and N. Vasconcellos, “Privacy preserving crowd monitoring: Counting people without people models or tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2008, pp. 1–7.

W. Li, L. Yongbo, and X. Xiangyang, “CODA: Counting objects via scale-aware adversarial density adaption,” in Proc. IEEE Int. Conf. Multimedia Expo (ICME), Jul. 2019, pp. 193–198.

M. A. Hossain, M. Kumar, M. Hosseinzadeh, O. Chanda, and Y. Wang, “One-shot scene-specific crowd counting,” in Proc. BMVC, 2019, p. 217.

J. Gao, Y. Yuan, and Q. Wang, “Feature-aware adaptation and density alignment for crowd counting in video surveillance,” IEEE Trans. Cybern., early access, Dec. 1, 2021, doi: 10.1109/TCYB.2020.3043416.

M. K. K. Reddy, M. A. Hossain, M. Rochan, and X. Wang, “Few-shot scene adaptive crowd counting using meta-learning,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2020, pp. 2814–2823.

M. Zhao, J. Zhang, C. Zhang, and W. Zhang, “Leveraging heterogeneous auxiliary tasks to assist crowd counting,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 12736–12745.

J. Wan, Q. Wang, and A. B. Chan, “Kernel-based density map generation for dense object counting,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Sep. 9, 2020, doi: 10.1109/TPAMI.2020.3022878.

H. Xiong, H. Lu, C. Liu, L. Liu, Z. Cao, and C. Shen, “From open set to closed set: Counting objects by spatial divide-and-conquer,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 8362–8371.

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

W. Kong, H. Li, G. Xing, and F. Zhao, “An automatic scale-adiaptive approach with attention mechanism-based crowd spatial information for crowd counting,” IEEE Access, vol. 7, pp. 66215–66225, 2019.

Z. Zou, Y. Cheng, X. Qu, S. 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.

Z. Zou, X. Qu, P. Zhou, S. Xu, X. Ye, W. Wu, and J. Ye, “Coarse to fine: Domain adaptive crowd counting via adversarial scoring network,” 2021, arXiv:2107.12858. [Online]. Available: http://arxiv.org/abs/2107.12858

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

Z. Wang, Z. Xiao, K. Xie, Q. Qiu, X. Zhen, and X. Cao, “In defense of single-column networks for crowd counting,” 2018, arXiv:1808.06133. [Online]. Available: http://arxiv.org/abs/1808.06133

Y. Xie, Y. Lu, and S. Wang, “RSANet: Deep recurrent scale-aware network for crowd counting,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Oct. 2020, pp. 1531–1535.

J. Chen, Q. Zhang, W.-S. Zheng, and X. Xie, “Efficient and switchable CNN for crowd counting based on embedded terminal,” IEEE Access, vol. 7, pp. 51533–51541, 2019.

N. Ilyas, A. Ahmad, and K. Kim, “CASA-Crowd: A context-aware scale aggregation CNN for crowd counting task,” IEEE Access, vol. 7, pp. 182050–182059, 2019.

Q. Wang, J. Gao, W. Lin, and X. Li, “NWPU-crowd: A large-scale benchmark for crowd counting and localization,” 2020, arXiv:2001.03560. [Online]. Available: http://arxiv.org/abs/2001.03560

L. Huang et al.: SRNet for Dense Crowd Counting

D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980

J. Gao, W. Lin, B. Zhao, D. Wang, C. Gao, and J. Wen, “C³ framework: An open-source PyTorch code for crowd counting,” 2019, arXiv:1907.02724. [Online]. Available: http://arxiv.org/abs/1907.02724

D. B. Sam, N. N. Sajjan, R. V. Babu, and M. Srinivasan, “Divide and grow: Capturing huge diversity in crowd images with incrementally growing CNN,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3618–3626.

M.-H. Oh, P. Olsen, and K. N. Ramamurthy, “Crowd counting with decomposed uncertainty,” in Proc. AAAI Conf. Artif. Intell., vol. 34, no. 7, 2020, pp. 11799–11806.

Y. Wang, J. Hou, X. Hou, and L.-P. Chau, “A self-training approach for point-supervised object detection and counting in crowds,” IEEE Trans. Image Process., vol. 30, pp. 2876–2887, 2021.

Y. Zhang, H. Zhao, F. Zhou, Q. Zhang, Y. Shi, and L. Liang, “MSCA: Adaptive multi-scale context aggregation network for congested crowd counting,” in Proc. Int. Conf. Multimedia Modeling, Cham, Switzerland: Springer, 2021, pp. 1–12.

Y. Hu, X. Jiang, X. Liu, B. Zhang, J. Han, X. Cao, and D. Doermann, “NAS-count: Counting-by-density with neural architecture search,” 2020, arXiv:2003.00217. [Online]. Available: http://arxiv.org/abs/2003.00217

X. Liu, J. Yang, and W. Ding, “Adaptive mixture regression network with local counting map for crowd counting,” 2020, arXiv:2005.05776. [Online]. Available: http://arxiv.org/abs/2005.05776

L. Liu, H. Lu, H. Zou, H. Xiong, Z. Cao, and C. Shen, “Weighing counts: Sequential crowd counting by reinforcement learning,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2020, pp. 164–181.

P. Thanasavesit, K.-I. Fukui, M. Numao, and B. Kijisirikul, “Encoder-decoder based convolutional neural networks with multi-scale-adaptive modules for crowd counting,” in Proc. 25th Int. Conf. Pattern Recognit. (ICPR), Jan. 2021, pp. 2382–2389.

A. Zhang, J. Shen, Z. Xiao, F. Zhu, X. Zhen, X. Cao, and L. Shao, “Relational attention network for crowd counting,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 6788–6797.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–775.

G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 4700–4708.

P. Hu and D. Ramanan, “Finding tiny faces,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 951–959.

C. Liu, X. Wei, and Y. Mu, “Recurrent attentive zooming for joint crowd counting and precise localization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 1217–1226.
LUNING ZHU was born in Shandong, China. He received the B.S. degree from Shandong Agricultural University, China, in 2019. He is currently pursuing the M.S. degree in control science and engineering with the School of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai, China. His research interests include image processing and deep learning.

SHIHUI SHEN was born in Anhui, China. He received the B.S. degree in software engineering from Daqing Normal University, Daqing, China, in 2019. He is currently pursuing the M.S. degree in resources and environment with Shanghai Institute of Technology, Shanghai, China. His research interests include crowd counting and image processing.

QING ZHANG received the B.S. and Ph.D. degrees from the School of Information Science and Engineering, East China University of Science and Technology, Shanghai, China, in 2007 and 2012, respectively. She is currently an Associate Professor with Shanghai Institute of Technology, Shanghai. Her research interests include image and video processing and computer vision.

JIANWEI ZHANG (Member, IEEE) received the B.Eng. (Hons.) and M.Eng. degrees from the Department of Computer Science, Tsinghua University, Beijing, China, in 1986 and 1989, respectively, the Ph.D. degree from the Institute of Real-Time Computer Systems and Robotics, Department of Computer Science, University of Karlsruhe, Karlsruhe, Germany, in 1994, and the Habilitation degree from the Faculty of Technology, University of Bielefeld, Bielefeld, Germany, in 2000. He is currently a Professor and the Head of the TAMS Group, Department of Informatics, University of Hamburg, Hamburg, Germany. He has been coordinating numerous collaborative research projects of the EU and German Research Council, including Transregio-SFB TRR 169 “Cross-modal Learning.” His current research interests include cognitive robotics, sensor fusion, dexterous manipulation, and multimodal robot learning. He is a Life-Long Academician of the Academy of Sciences, Hamburg. He is the General Chair of the IEEE MFI 2012, the IEEE/RSJ IROS 2015, and the IEEE Robotics and Automation Society AdCom, from 2013 to 2015.