Multi-level feature fusion network for crowd counting

Luyang Wang\(^1\) | Yun Li\(^1\) | Sifan Peng\(^1\) | Xiao Tang\(^2\) | Baoqun Yin\(^1\)

\(^1\)Department of Automation, University of Science and Technology of China, Hefei, China
\(^2\)Department of Electronic Science and Technology, University of Science and Technology of China, Hefei, China

Abstract
Crowd counting has become a noteworthy vision task due to the needs of numerous practical applications, but it remains challenging. State-of-the-art methods generally estimate the density map of the crowd image with the high-level semantic features of various deep convolutional networks. However, the absence of low-level spatial information may result in counting errors in the local details of the density map. To this end, a novel framework named Multi-level Feature Fusion Network (MFFN) for single image crowd counting is proposed. The proposed MFFN, which is constructed in an encoder-decoder fashion, incorporates semantic and spatial information for generating high-resolution density maps of input crowd images. Skip connections are developed between the encoder and the decoder so that low-level spatial information and high-level semantic features can be combined by element-wise addition. In addition, a dense dilated convolution block is placed behind the encoder, extracting multi-scale context features to guide feature fusion by a channel attention mechanism. The model is trained by multi-task learning; semantic segmentation supervision is introduced to enhance feature representation. Extensive experiments are conducted on three crowd counting datasets (ShanghaiTech, UCF_CC_50, UCF-QNRF), and the results show that MFFN outperforms state-of-the-art methods. In addition, sufficient ablation studies are performed to verify the effectiveness of each component in our proposed method.

1 | INTRODUCTION
The rapid growth of the urban population has made massive crowd gatherings commonplace such as sports, concerts and subway stations. In recent years, stampedes have occurred frequently around the world, causing tremendous damage to human life and property, which reminds us of the importance of crowd control. Crowd counting, which aims to precisely estimate the number of people in an image, is essential for crowd control and public safety in crowded scenes. In addition, accurate crowd estimates can also play an important role in journalism and urban planning. However, it remains a challenge due to various issues such as occlusions, non-uniform distribution, scale variations and perspective. Figure 1 shows some representative images reflecting the above problems.

Early solutions for crowd counting rely on detection [1–3] and regression [4–6] with hand-crafted features, and limited feature representation capabilities restrict the increase in counting performance. Recently, numerous CNN-based methods for crowd counting have been proposed to estimate the density map of the crowd image. The integral of the density map denotes the global count of the crowd image. Most of these methods [7–9] focus on multi-column architectures to address large variations of crowd sizes in an image. Multiple CNN columns with receptive fields of different sizes are designed in these architectures to extract features at different scales. More recently, researchers have further improved the performance of crowd counting by deepening the network to enlarge the receptive field [10] or generating high-resolution density maps [11,12]. Compared with previous works using low-resolution density maps, the intuitive advantage is that the high-resolution density map contains more subtle details that contribute to crowd density estimation.

However, inherent drawbacks exist in these methods. On the one hand, multi-column networks consume a lot of training time, computing and memory resources, and the limited number of columns restricts the scale diversity of features. Furthermore, different columns in a multi-column architecture learn similar features, which may result in ineffective branch structures [10]. On the other hand, all of the...
above methods generate density maps using only high-level semantic features, and none of them take into account spatial details. Recent solutions typically leverage the backbone of classification networks to extract crowd features and then stack layers to transform the crowd features into the density map. As the network grows deeper, the spatial detail information contained in the low-level of the network gradually disappears, and the high-level features are occupied by semantic information due to the increase in the size of the receptive field. The lack of spatial information may lead to inaccurate crowd density estimation in local details.

To cope with the above issues, we introduce an encoder–decoder architecture called Multi-level Feature Fusion Network (MFFN) to fuse semantic and spatial information and generate high-resolution density maps, as shown in Figure 2. The high-resolution density map is designed as the regression objective, since it can describe crowds with more pixel points to retain more accurate details of small objects. The encoder–decoder network can output maps with the same size as inputs. It balances feature extraction and spatial resolution preservation by learning a pixel-wise mapping. In MFFN, we divide the first ten layers of VGG-16 [13] into four blocks as the encoder and deploy a set of convolutions and upsampling layers as the decoder. At the meanwhile, we hold the view that crowd density estimation as a low-level vision task is different from classification. For density map estimation, only the semantic feature is not enough, and spatial details are also indispensable. Spatial information can compensate for the limited ability of semantic information to describe local details in density maps. Therefore, we fuse semantic and spatial information in MFFN to promote crowd density regression.

As mentioned above, the features of different layers in the network have different levels of abstraction. A natural idea is to combine low-level and high-level features together to fuse semantic and spatial information. Inspired by FPN [14], we build skip connections between the encoder and the decoder by element-wise addition. However, we believe that not all fused features can facilitate crowd density map estimation, due to the diverse consistency of different stages and the lack of global context during feature fusion. While skip connections enhance multi-level feature fusion, the fact that some fused features may have a negative impact is neglected. Based on this consideration, it is desirable to adjust the weight of fused features at every stage by consulting global context. A dense dilated convolution block (DDCB) is designed to extract global context features, which contains four densely connected dilated convolutional layers behind the encoder. Extra semantic segmentation supervision is introduced to classify pixels into backgrounds and crowds for auxiliary learning. Then, we adjust the weight of the feature channel at each fusion stage under the guidance of DDCB context features. We recalibrate the fused feature responses using a channel attention module (CAM) to highlight the channels that facilitate performance improvements. The main difference between our MFFN and FPN is that we refine the fusion process by introducing fused feature adjustments on the base of feature fusion.

The full framework of our MFFN is trained in a multi-task learning fashion. Different from some other works using crowd density level classification as an auxiliary task, we integrate semantic segmentation supervision during training to make the context feature more distinguishable. In summary, the main contributions are as follows:

FIGURE 1 Examples of crowd scenes in the ShanghaiTech dataset. The large variations in background, perspective, crowd distribution and pedestrian size limit the performance of crowd counting.
We propose a novel encoder-decoder network called Multi-level Feature Fusion Network to incorporate spatial and semantic information for generating high-resolution crowd density maps.

- We improve the fusion process at every stage by fused feature channel adjustment. A dense dilated convolution block in combination with a channel attention module is designed to recalibrate the weight of the fused feature channel in accordance with global context.
- Extensive experiments conducted on three crowd counting datasets demonstrate that our MFFN achieves superior performance compared with recent state-of-the-art methods.

The rest is organized as follows. We first briefly introduce the related work of crowd counting in Section 2. Section 3 describes the architecture of our proposed MFFN in detail, and Section 4 shows the experimental results. Finally, we conclude in Section 5.

2 | RELATED WORK

A large number of methods have been proposed for crowd counting, and they can be roughly summarized into two categories: traditional methods and CNN-based methods. Early traditional approaches focus on either detection or regression using hand-crafted features. CNN-based approaches significantly improve counting performance due to the powerful feature representation capabilities of CNNs. We briefly review several representative methods in each category, and a more comprehensive review of crowd counting can be found in previous studies [15–18].

2.1 | Traditional methods

Early solutions [2,3,19–21] for crowd counting generally tend to detect pedestrians in an image. These methods typically train a head or body detector to locate people and sum detected individuals up as the final count. Lin and Davis [22] matched a part-template tree to images hierarchically to detect humans and estimate their poses. Wu and Nevatia [2] trained a human body part detector by a boosting method based on edgelet features. However, detection-based methods suffer from occlusions and clutters in highly congested scenes. Obscured pedestrians and small-scale individuals in dense crowd scenes are difficult to detect.

To address the above issues, researchers turn their attention to regression-based approaches [23–27], where they learn a mapping from the features of local patches to their counts. Feature extraction and regression modelling are two main components of these methods. On this base, Idriss et al. [28] estimated crowd counts by incorporating features of multiple sources, such as Fourier analysis, head detection and SIFT [29] features. Chen et al. [5] proposed a multi-output regression model to discover the inherent importance of different features for crowd counting. Although these methods perform well for dealing with occlusions and clutters, only the final count is provided and the crowd spatial distribution information is ignored. Lemptisky et al. [30] introduced a method for counting tasks to learn a linear mapping from features to object density maps. The density map depicts the distribution of crowds, and its integral gives the global count. Pham et al. [31] learned a nonlinear mapping using random forest regression for estimating density maps. The density map is also widely used as the regression objective in CNN-based methods, since it can balance the descriptions of spatial distribution and the global count. All of the above approaches use hand-crafted features, which is the main difference from CNN-based methods.

2.2 | CNN-based methods

Inspired by the success of deep learning in various vision tasks such as image classification [32], semantic segmentation [33,34] and object detection [35], a large number of CNN-based
approaches have also been proposed to address crowd counting and achieve performance improvements. In the study by Wang et al. [36], the authors modified AlexNet [37] to directly predict the number of people in a crowd image. Shang et al. [38] designed an end-to-end CNN to estimate the global and local counts of crowd images, simultaneously. In contrast, more methods employ CNNs to estimate crowd density maps. Zhang et al. [39] incorporated perspective information into the density map and proposed an alternately trained CNN for cross-scene crowd counting. However, the use of perspective maps limits the availability of this method in practical applications. Huang et al. [40] formulated the human body, head and context structure as semantic models and converted the crowd counting task into a multi-task learning problem.

To address the issue of scale variations, multi-column networks are introduced to cover diverse pedestrian scales. Zhang et al. [7] proposed Multi-column CNN (MCNN) containing three convolution columns with receptive fields of different sizes. A similar idea was depicted in the study by Onoro-Rubio and López-Sastre [41], where the authors introduced a model with multi-scale inputs to extract features at different scales. Inspired by MCNN, Sam et al. [8] found that generating density maps by late fusion of features from multiple columns may degrade performance. They instead introduced Switch-CNN that uses a density level classifier to select the optimal regressor for a particular input patch. Sindagi et al. [9] incorporated global and local density information into a multi-column network for generating high-quality density maps. Multiple CNN columns result in more parameters to be optimized and consume more memory and computing resources. Due to the lack of explicit supervision, different convolutional branches are difficult to learn distinguishable features, which is contrary to the original design intent.

Recently, deeper network structures and the use of high-resolution density maps further improve the performance of crowd counting. In the study by Li et al. [10], a deep single-column network referred to CSRNet was presented, which uses dilated convolutions to enlarge the receptive field and extract deep features. Cao et al. [11] proposed an encoder-decoder structure named SANet for generating high-resolution density maps. Wu et al. [42] presented an adaptive scenario discovery framework with two deep pathways for modelling the dynamic scenarios implicitly. In the study by Babu Sam et al. [43], the authors tackled crowd counting with a growing CNN which can progressively increase its capacity to account for the wide variability seen in crowd scenes. In [44], a deep detection network with only point supervision required is proposed. It can simultaneously detect the size and location of human heads and count them in crowds. Gao et al. [45] designed a multi-task counting architecture, which simultaneously performs density map estimation, high-level density classification and foreground segmentation. Liu et al. [46] introduced an attention-injective deformable convolutional network. It first detects crowd regions in images and then generates crowd density maps. Zeng et al. [47] proposed a deep-scale purifier network that can encode multi-scale features and reduce the loss of contextual information for dense crowd counting. However, all of these methods fail to consider including spatial details into crowd density map estimation. Based on the above observations, we propose an encoder-decoder network to incorporate spatial details and semantic features for generating high-resolution and high-quality crowd density maps.

3 | PROPOSED METHOD

A detailed description of MFFN is presented in this section. We first introduce the overall network architecture illustrated in Figure 2, and the setting of each module, and then give the training details.

3.1 | Architecture

As aforementioned, the multi-column architecture for crowd counting has inherent defects, and existing methods only use the high-level features of the network to estimate crowd density maps. The higher layers of the network mainly contain semantic information, since the spatial details are gradually lost as the network deepens. The generation of density maps relying solely on semantic information leads to estimation errors in local details, as the density map directly describes the crowd spatial distribution. Moreover, most of these methods generate low-resolution density maps due to the presence of pooling layers, which may also lose spatial details. Therefore, the main idea of the proposed MFFN is to incorporate high-level semantic features and low-level spatial features and generate the density map with the same size as the original image.

The backbone of MFFN is an encoder-decoder structure, as shown in Figure 2. Following previous works [10,48], we choose the first ten convolutional layers of VGG-16 as the encoder because of its powerful transfer learning capabilities and appropriate depth. The ten layers are divided into four blocks according to the locations of the three pooling layers. These four blocks have different feature representation capabilities and receptive field sizes. The encoder takes the crowd image as input, and its output feature maps are directly input to the decoder. The decoder transforms high-dimensional features into the high-resolution density map by a set of convolutional layers and upsampling operations. Two convolutional layers are first used to refine the feature maps, and then bilinear interpolation is performed for upsampling. To incorporate low-level and high-level features, we build skip connections between the encoder and the decoder. After a convolution, the features of Block 3 in the encoder are fused with the upsampled features in the decoder by element-wise addition. Although the fused features merge spatial details and semantic components, not all of them are helpful for crowd density estimation due to the lack of global context information. To emphasize informative feature maps and suppress
useless ones, the channel-wise feature responses are recalibrated by the collaboration between two modules we designed, DDCB and CAM. The DDCB placed behind the encoder is used to extract the global multi-scale context with extra semantic segmentation supervision. CAM leverages the global context that DDCB outputs as a guide to assign weights to each channel of the fused feature. The same operations are performed one more time to fuse the features of Block 2, and the fused features are also recalibrated by CAM. After that, the size of the feature map is further increased to the same size as the input image through a series of convolutions and an upsampling layer. The features of Block 1 are not chosen for feature fusion, considering that only two convolutions are conducted on the input image. Finally, the decoder uses a $1 \times 1$ convolutional layer to generate the high-resolution density map.

The backbone of the proposed encoder–decoder network fuses multi-level features by skip connections, which is closely related to FPN [14] for object detection. Both two models combine high-level semantic features and low-level spatial features using element-wise addition. However, this operation just simply sums features up, which ignores the diverse consistency of features at different stages. The fused features are employed directly by subsequent layers or used for prediction, but lack global multi-scale context. Therefore, the main difference between FPN and our MFFN is that we improve the fusion process by adjusting fused feature channels as mentioned above. For crowd counting, this design uses crowd context to promote the screening of fused features, and experimental results prove its effectiveness in Section 4.5.

In addition to high-resolution density map estimation as the major training objective, a semantic segmentation task is also introduced for auxiliary context feature learning. The crowd image in actual application scenarios inevitably contains various complex backgrounds. Some background parts are similar to the congested crowds leading to incorrect estimation of crowd density, since a single pedestrian in the dense crowd only occupies limited pixels. Semantic segmentation supervision can force the model to focus on the crowd areas and provide crowd location information, thereby promoting crowd density map estimation. Therefore, pixel-wise classification is employed as an auxiliary task in our proposed MFFN. As shown in Figure 2, a convolution layer behind DDCB classifies pixels into two categories: backgrounds and crowds, which makes DDCB features distinguishable and further improves feature fusion. Annotating foreground may consume a lot of labour and resources, we therefore generate the ground truth label map of semantic segmentation by density map binarization. Pixels that are not equal to zero in the density map belong to crowds and others are backgrounds. Although the segmentation label is relatively coarse, accurate classification for each object is not critical. As an auxiliary, it is enough to provide attentive information of crowds for density map estimation. Previous works use density-level classification as an auxiliary task and only employ the rough global density feature, while the segmentation task forces the model to learn discriminative features for the distinction between backgrounds and crowds. Experimental results show that these discriminative features contribute more to density map estimation.

### 3.2 Dense dilated convolution block

As shown in Figure 1, people in unconstrained scenes exhibit large variations posing great challenges for recognition, and therefore discriminative feature extraction is crucial. To enhance feature learning, we propose a dense DDCB to exploit discriminative context features. DDCB consists of a set of densely connected dilated convolutional layers, which not only increases the size of the receptive field but also covers a wide range of scales. Li et al. [10] first introduced the dilated convolution for crowd counting, because it is able to achieve a larger receptive field without losing the size of the feature map by inserting zeros into the convolution kernel. However, we believe that only a large receptive field is not enough to extract discriminative context features. Motivated by DenseNet [49], we therefore consider connecting dilated convolutional layers in a dense way.

Figure 3 depicts the structure of DDCB containing four dilated convolutional layers and a traditional $1 \times 1$ convolutional layer. The dilated convolution is formulated as follows:

$$y(l, m) = \sum_{i=1}^{L} \sum_{j=1}^{M} x(l + r \cdot i, m + r \cdot j) w(i, j),$$

where $x$ and $y$ are the input and output of the dilated convolution, respectively. $w$ denotes a filter with the length of $L$ and the width of $M$. $r$ is the dilation rate. To simplify notations, we use $D_{k, r}(x)$ to represent a dilated convolution with a filter of $k \times k$ and a dilation rate of $r$. Consequently, the densely connected dilated convolution in our DDCB can be written as:

$$x_l = D_{k, r}(\left[ x_0, \ldots, x_{l-1} \right]),$$

where $x_l$ means the $l^{th}$ output features, and $\left[ x_0, \ldots, x_{l-1} \right]$ is the concatenation of the features in 0, ..., $l - 1$ layers. DDCB stacks four dilated convolutional layers with $256 \times 3 \times 3$ filters and dilation rates of (2, 2, 4, 4). A $1 \times 1$ convolutional layer can reduce the number of output feature maps. Finally, the output features $y$ of DDCB can be formulated as:

**FIGURE 3** The architecture of the dense dilated convolution block (DDCB). $d$ denotes the dilation rate of a convolutional layer
The proposed DDCB aims to extract multi-scale context features and guide fused feature adjustments for dealing with various variations in crowd counting rather than simply combining dense connections and dilated convolutions. In [50], the authors demonstrate through experiments that extracting multi-scale features at a relatively late stage of the network is more efficient than at early layers. DDCB located behind the encoder densely concatenates high-level features with large respective fields, which can facilitate multi-scale context representation. Different from DenseNet which aims to deepen the network, DDCB increases receptive fields of different sizes by dense connections and covers a wide scale range of objects. Dense connections in DDCB mainly play the role of extracting multi-scale features and coping with scale variations.

### 3.3 Channel attention module

The crowd features at different stages retain inconsistencies, which influences the fused feature representation. To improve the representation ability of fused features, a CAM is introduced to adjust the weights of fused features at each stage, as illustrated in Figure 4. According to the context features of DDCB, CAM adaptively assigns larger weights to the feature channels that contribute to density estimation and suppresses the useless feature channels.

The output features of DDCB are first fed into a global average pooling layer to aggregate spatial information and produce a channel descriptor embedded the global context. The descriptor $z_c$ of the $c^{th}$ channel is generated by shrinking spatial dimensions $H \times W$, which can be calculated by:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \gamma_c(i,j).$$

It is followed by two $1 \times 1$ convolutional layers to capture channel-wise dependencies. The channel attention is defined as:

$$s = \sigma(W_2 \delta(W_1 z)),$$

where $\delta$ is the ReLU function, and $\sigma$ means the sigmoid activation. $W_1$ and $W_2$ denote two convolutions in CAM.

The final output $\tilde{f}$ of the module is obtained by zooming the fused features $f_c$ as formulated in Equation (6).

A similar technique referred to SENet [51] is proposed to model interdependencies between channels. SENet learns the channel attention of a layer only based on the features of that layer, therefore its context information is limited. Our CAM, however, uses the global context derived from DDCB to guide channel attention learning for promoting more accurate adjustments. The multi-scale context features extracted by DDCB provides more diverse feature representation compared with single layer features.

### 3.4 Training details

The proposed network is trained in an end-to-end manner with two supervisions. We introduce the training details about the ground truth and loss function in this section.

#### 3.4.1 Ground truth

For density map estimation, we generate the ground truth following the method in [7]. The geometry-adaptive Gaussian kernel normalized to one is used to blur head annotations, and the ground truth density map can be defined as follows:

$$F(x) = \sum_{i=1}^{N_h} \delta(x - x_i)^{2} G_{\sigma_i}(x), \quad \sigma_i = \beta \times \overline{d}_i,$$

where $x_i$ is the two-dimensional coordinate of the $i^{th}$ head annotation, and it is represented with a delta function $\delta(x - x_i)$. $N_h$ means the total number of head annotations. The parameter $\sigma_i$ that equals $\beta \times \overline{d}_i$ denotes the standard deviation of a Gaussian kernel, in which $\overline{d}_i$ is the average distance of $k$ nearest neighbours. In our experiments, we set $\beta = 0.3$ and $k = 5$.

The ground truth segmentation label is generated based on the density map due to the lack of precise annotations. The pixels are roughly classified into the background and the crowd. The segmentation label is rough but effective, since we focus on density map estimation and mainly employ segmentation for auxiliary learning. It can be defined as follows,

**Figure 4** The architecture of the channel attention module (CAM)
where $D[\cdot]$ denotes 8 times down-sampling, and $F(x)$ means the ground truth density map. 1 and 0 represent crowds and backgrounds, respectively. Pixels that are not equal to 0 in the density map correspond to the crowd category of the segmentation label. In contrast, the background of segmentation is the point with a value of 0 in the density map.

### 3.4.2 Loss function 1

The pixel-wise Euclidean loss is used to measure the difference between the estimated density map and the corresponding ground truth, which is formulated as follows:

$$L_E = \frac{1}{N_p} \| F(X; \Theta) - F \|^2_2,$$

where $N_p$ is the number of pixels, and $\Theta$ is the parameters to be optimized. $F$ and $F(X; \Theta)$ are the ground truth and estimated density map, respectively. As we know, the Euclidean loss is sensitive to outliers and leads to blurred results.

Therefore, another attempt aims at minimizing the SSIM loss to overcome these issues and improve the quality of results. The SSIM loss measures the local pattern consistency of the ground truth and estimated density map in the study by Cao et al. [11], which is defined below:

$$L_S = 1 - \frac{1}{N_p} \sum_x \text{SSIM}(x).$$

For the semantic segmentation task, we choose the cross entropy loss as the loss function,

$$L_C = \frac{1}{N_p} \sum_{i=1}^{N_p} [t_i \times \log(y_i) + (1 - t_i) \times \log(1 - y_i)],$$

where $t_i$ and $y_i$ are the ground truth and estimated values, respectively. The overall loss $L$ is calculated by weighting three loss functions,

$$L = L_E + aL_S + \beta L_C,$$

where $a$ and $\beta$ are hyperparameters, and we empirically set $a = 10^{-3}$ and $\beta = 10^{-5}$.

# 4 EXPERIMENTS

We evaluate the performance of our MFFN on three crowd counting datasets including ShanghaiTech [7], UCF_CC_50 [28] and UCF-QNRF [52]. Data augmentation is conducted to increase the number of training samples as well as avoid overfitting. We crop nine patches for each training image and flip them horizontally so that each image produces 18 training samples. The size of each patch is 1/4 of the original image. The encoder in MFFN is initialized with the parameters of a well-trained VGG-16 on ImageNet. The initial values of other layers follow a Gaussian distribution with a standard deviation of 0.01. The Adam [53] optimizer with a learning rate of $10^{-5}$ is applied to train the model. All the experiments are based on the Pytorch [54] framework. In this section, we first introduce the evaluation metrics, and then compare the proposed network with previous state-of-the-art methods on three datasets. Finally, ablation studies report the effectiveness of each module in MFFN.

## 4.1 Evaluation metrics

Following previous works [7,11,55], we choose the mean absolute error (MAE) and the mean squared error (MSE) as evaluation metrics. They are defined as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|,$$

$$MSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2},$$

where $N$ is the number of test images, $y_i$ and $\hat{y}_i$ are the ground truth and estimated crowd count of the $i^{th}$ image, respectively. MAE denotes the accuracy of the estimated count, and MSE indicates robustness.

## 4.2 ShanghaiTech dataset

The ShanghaiTech dataset is first introduced in the study by Zhang et al. [7] and consists of 1198 annotated crowd images with a total of 330,165 people. It is divided into two parts: Part_A and Part_B. Part_A has 482 images collected from the Internet, of which 300 are training images and the remaining 182 are for testing. Part_B includes 716 images taken from the busy streets of Shanghai. The training and test sets of Part_B contain 400 and 316 images, respectively. The scenes of Part_A are highly congested, while the images in Part_B are relatively sparse with large-scale variations. Figure 5 shows examples of ground truth and estimated density maps in the ShanghaiTech dataset.

We compare the count error of MFFN on the ShanghaiTech dataset with 11 recent state-of-the-art methods, as shown in Table 1. All of these methods are CNN-based approaches proposed in recent years which have delivered superior results. Our proposed method obtains the lowest MAE and MSE among all the methods on both Part_A and Part_B. Specially, MFFN achieves an improvement of 4.3% for MAE in comparison to the second best solution RAZ-Net on Part_A. It indicates that MFFN is effective in both congested and sparse scenes.

To further evaluate the count errors of our proposed method at different density levels, we follow the setting of
We perform a fivefold cross-validation to evaluate the performance of the proposed method, following the standard setting in the study by Idrees et al. [28]. As tabulated in Table 2, our method achieves the best MAE and a competitive MSE compared with other recent methods. The MAE of MFFN is 10.1% lower than that of the second best method TEDNet. The results denote that our MFFN not only performs well in relatively sparse scenes but also works in dense scenarios. Moreover, it also indicates that the fusion of spatial details and semantic information is effective for count error reduction, since many objects in UCF_CC_50 are quite dense and only occupy a few pixels.

We also compare the estimated count with the ground truth for each image in UCF_CC_50, as shown in Figure 8. Most points in the estimated line are close to their actual counts, while several images with extremely large crowd counts present obvious estimation errors. We think this is due to the fact that the proportion of extremely dense images in the dataset is too small.

### 4.4 UCF-QNRF dataset

To the best of our knowledge, UCF-QNRF [52] is currently the largest crowd counting dataset in terms of the number of people annotated. This dataset consists of 1535 crowd images of different resolutions with 1,251,642 people. There are 1201 and 334 images in the training set and test set, respectively. The number of people in an image ranges from 49 to 12,865, whereas the median and mean counts are 425 and 815. Figure 9 shows the density maps estimated by MFFN and their ground truth maps in the UCF-QNRF testset. Although the scenes and scales in the dataset vary greatly, our MFFN can produce results that approximate the ground truth density maps.

Table 3 lists the results of different approaches including our MFFN. The results show that our proposed method outperforms all of the previous methods in MAE and attains a comparable MSE. MFFN demonstrates a reduction of eight over the second best method on the MAE metric.

---

**FIGURE 5** Density map visualization of the proposed MFFN on the ShanghaiTech dataset. The left three columns and right three columns are examples in Part_A and Part_B, respectively.

**TABLE 1** Count errors of different methods on ShanghaiTech

| Method            | Part_A |      | Part_B |      |
|-------------------|--------|------|--------|------|
|                   | MAE    | MSE  | MAE    | MSE  |
| Zhang et al. [39] | 181.8  | 277.7| 32.0   | 49.8 |
| MCNN [7]          | 110.2  | 173.2| 26.4   | 41.3 |
| Switch-CNN [8]    | 90.4   | 135.0| 21.6   | 33.4 |
| CP-CNN [9]        | 73.6   | 106.4| 20.1   | 30.1 |
| Shi et al. [56]   | 107.6  | 169.3| 21.4   | 33.9 |
| Liu et al. [57]   | 72.0   | 106.4| 14.4   | 23.8 |
| D-ConvNet-v1 [48] | 73.5   | 112.3| 18.7   | 26.0 |
| CSRNet [10]       | 68.2   | 115.0| 10.6   | 16.0 |
| SANet [11]        | 67.0   | 104.5| 8.4    | 13.6 |
| PSDNN [44]        | 65.9   | 112.3| 9.1    | 14.2 |
| RAZ-Net [58]      | 65.1   | 106.7| 8.4    | 14.1 |
| MFFN (ours)       | 62.3   | 104.2| 8.3    | 13.0 |

this dataset and divide the test images into 10 groups according to the number of people in the image. The first nine groups have 18 and 31 test images in Part_A and Part_B, and there are 20 + 37 images in the last group. Figure 6 depicts the ground truth and estimated counts at different density levels. On both Part_A and Part_B, the lines of the ground truth and MFFN are very close and almost coincident, which proves that our method can achieve small count errors at every density level.

### 4.3 UCF_CC_50 dataset

The UCF_CC_50 dataset [28] is extremely challenging, since only 50 images with head annotations are provided. The crowd count of each image in this dataset varies widely, ranging from 94 to 4543. A total of 63,874 annotations are marked with an average of 1280 per image. Figure 7 shows examples in UCF_CC_50 and the visual comparison of the ground truth and estimated density maps.
Ablation experiments

We conduct various ablation studies to analyse the impact of different configurations of our method. ShanghaiTech Part_A is chosen for the experiment due to its diversity in crowd density and scales. We compare the MAE and MSE metrics of different settings from three aspects including loss function, dilation rates of DDCB and the configuration of architecture.
We use the combination of three loss functions defined in Equation (12) to train the model, where the pixel-wise Euclidean loss and the SSIM loss are introduced to supervise density map estimation, and the cross entropy loss is for the semantic segmentation task. To verify the effectiveness of each loss, the three loss functions are added to the network training one by one, and the evaluation results are summarized in Table 4. The results show that training the model with the Euclidean loss and the SSIM loss together can reduce the MAE metric from 65.7 to 63.6 compared with using only the Euclidean loss. When the three loss functions are used simultaneously, the count error can be further reduced to 62.3 in MAE, and the MSE metric also decreases from 110.9 to 104.2. This experiment demonstrates that the auxiliary supervision of semantic segmentation contributes to density map estimation.

4.5.2 | Dilation rates of DDCB

DDCB is designed to boost context feature learning in our proposed MFFN, which contains four densely connected dilated convolutional layers as mentioned in Section 3.2. We compare the count errors of three configurations in terms of the dilation rate, as reported in Table 5. Two of the three configurations are modules with the dilation rate of 2 or 4. The last module combines the dilation rate of 2 and 4, and it achieves the lowest MAE and MSE. Then we choose the last configuration in all experiments.

4.5.3 | Architecture

We also investigate the effect of each module in the proposed method and train four different architectures. The performance of each architecture is depicted in Table 6. W/O DDCB + CAM means removing DDCB and CAM from MFFN. The output of the encoder is only employed by the decoder, and the fused features of each stage in the decoder are directly used by the next stage without weight adjustment. W/O CAM denotes that CAM is removed, and the decoder takes as input the features of DDCB instead of the output of the encoder. W/O DDCB indicates that MFFN removes DDCB, and CAM takes the features of the encoder as input instead. Finally, MFFN represents the entire network of the proposed method. It can be observed that the network with DDCB and CAM can achieve better counting performance. Moreover, the simultaneous use of DDCB and CAM achieves the best MAE and MSE. It demonstrates that each part of MFFN is effective in reducing the count error.

5 | CONCLUSION

Herein, we propose a novel encoder–decoder network named MFFN to incorporate spatial details and semantic information for generating high-resolution crowd density maps. The first ten layers of VGG-16 are used as the encoder, and the decoder containing a set of convolutional layers and upsampling operations generates the final density map. The low-level spatial features and high-level semantic features are combined by skip connections between the encoder and the decoder. In addition, we improve the fusion process to deal with the diverse consistency at different stages. A dense DDCB extracts context features to guide channel-wise fused feature adjustment by a CAM. The overall architecture is trained by multi-task learning, foreground segmentation as an auxiliary task promotes crowd density map estimation. Extensive experiments on three crowd counting datasets are conducted to compare with state-of-the-art methods, and our proposed method outperforms all recent approaches.
**Figure 9** Density map visualization of the proposed MFFN on the UCF-QNRF dataset. First column: test images. Second column: ground truth density maps. Third column: estimated density maps.

| Table 3 | Count errors of different methods on UCF-QNRF |
|---------|---------------------------------------------|
| Method             | MAE  | MSE  |
| Idrees et al. [28] | 315  | 508  |
| MCNN [7]           | 277  | 426  |
| Switch-CNN [8]     | 228  | 445  |
| ResNet101          | 190  | 277  |
| DenseNet201        | 163  | 226  |
| Idrees et al. [52] | 132  | 191  |
| MFFN (ours)        | **124** | **227** |

| Table 4 | Estimation errors of MFFN with different loss functions |
|---------|----------------------------------------------------------|
| Loss Function | MAE | MSE |
| $L_E$       | 65.7 | 110.9 |
| $L_E, L_S$  | 63.6 | 110.9 |
| $L_E, L_S, L_C$ | **62.3** | **104.2** |

| Table 5 | Comparison of DDCB with different dilation rates |
|---------|--------------------------------------------------|
| Model            | MAE  | MSE  |
| DDCB(2,2,2,2)    | 64.4 | 114.5 |
| DDCB(4,4,4,4)    | 63.9 | 109.7 |
| DDCB(2,2,4,4)    | **62.3** | **104.2** |

| Table 6 | Estimation errors of different network configurations |
|---------|----------------------------------------------------------|
| Model            | MAE  | MSE  |
| W/O DDCB + CAM   | 66.9 | 116.8 |
| W/O CAM          | 64.5 | 111.4 |
| W/O DDCB        | 66.1 | 108.2 |
| MFFN             | **62.3** | **104.2** |
REFERENCES

1. Ge, W., Collins, R.T.: Marked point processes for crowd counting. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2913–2920, (2009)

2. Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: Proceeding of International Conference on Computer Vision, pp. 90–97, (2005)

3. Wang, M., Wang, X.: Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3401–3408, (2011)

4. Chan, A.B., Vasconcelos, N.: Bayesian Poisson regression for crowd counting. In: Proceedings of International Conference on Computer Vision, pp. 545–551, (2009)

5. Chen, K., et al.: Feature mining for localised crowd counting. In: Proceedings of the British Machine Vision Conference, (2012)

6. Paragios, N., Ramesh, V.: A mrf-based approach for real-time subway monitoring. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition 1–I, (2001)

7. Zhang, Y., et al.: Single-image crowd counting via multi-column convolutional neural network. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 589–597, (2016)

8. Sam, IAB, Surya, S., Babu, R.V.: Switching convolutional neural network for crowd counting. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4031–4039, (2017)

9. Sindagi, V.A., Patel, V.M.: Generating high-quality crowd density maps using contextual pyramid cnns. In: Proceeding of the IEEE Conference on Computer Vision, pp. 1879–1888, (2017)

10. Li, Y., Zhang, X., Chen, D.: Crsnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1091–1100, (2018)

11. Cao, X., et al.: Scale aggregation network for accurate and efficient crowd counting. In Proceedings of the European Conference on Computer Vision, pp. 757–773, (2018)

12. Ranjan, V., Le, H., Hoai, M.: Iterative crowd counting. In Proceedings of the European Conference on Computer Vision, pp. 270–285, (2018)

13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. (2014)

14. Lin, T.-Y., et al.: Feature pyramid networks for object detection. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125, (2017)

15. Sindagi, V.A., Patel, V.M.: A survey of recent advances in cnn-based single image crowd counting and density estimation. Pattern Recogn. Lett. 107, 3–16, (2018)

16. Zitouni, M.S., et al.: Advances and trends in visual crowd analysis: a systematic survey and evaluation of crowd modelling techniques. Neurocomputing, 186, 139–159 (2016)

17. Li, T., et al.: Crowded scene analysis: a survey. IEEE Trans. Circuits Syst. Video Technol. 25(3), 367–386 (2015)

18. Teixeira, T., Duhlon, G., Savvides, A.: A survey of human-sensing methods for detecting presence, count, location, track, and identity. ACM Comput. Surv. 51(1), 59–69 (2010)

19. Lin, S.-F., Chen, J-Y., Chao, H.-X.: Estimation of number of people in crowded scenes using perspective transformation. IEEE Trans. Man. Cybern. A. 31(6), 645–654 (2001)

20. Dollar, P., et al.: Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. 34(4), 743–761 (2012)

21. Li, M., et al.: Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: Proceeding of the International Conference on Pattern Recognition, pp. 1–4, (2008)

22. Lin, Z., Davis, L. S.: Shape-based human detection and segmentation via hierarchical part-template matching. IEEE Trans Pattern. Anal. Mach. Intell. 32(4), 604–618 (2010)

23. Xu, B., Qiu, G.: Crowd density estimation based on rich features and random projection forest. IEEE Winter Conference on Applications of Computer, pp. 1–8, (2016)

24. Chen, K., et al.: Cumulative attribute space for age and crowd density estimation. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2467–2474, (2013)

25. Ryan, D., et al.: Crowd counting using multiple local features. Proceedings of the Digital Image Computing: Techniques and Applications, pp. 81–88, (2009)

26. Marana, A., et al.: On the efficacy of texture analysis for crowd monitoring. In: Proceedings of the International Symposium on Computer Graphics Image Processing and Vision, pp. 354–361, (1998)

27. Kong, D., Gray, D., Tao, H.: A viewpoint invariant approach for crowd counting. In: Proceedings of the International Conference on Pattern Recognition. Vol. 3, pp. 1187–1190, (2006)

28. Idrees, H., et al.: Multi-source multi-scale counting in extremely dense crowd images. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2547–2554, (2013)

29. Lowe, D.G., et al.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision, 1150–1157 (1999)

30. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: Proceedings of the International Conference on Neural Information Processing, pp. 1324–1332, (2010)

31. Pham, V-Q., et al.: Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In: Proceedings of the International Conference on Computer Vision, pp. 3253–3261, (2015)

32. He, K., et al.: Deep residual learning for image recognition. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, (2016)

33. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440, (2015)

34. Yu, C., et al.: Learning a discriminative feature network for semantic segmentation. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1857–1866, (2018)

35. Ren, S., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings International Conference on Neural Information Processing, pp. 91–99, (2015)

36. Wang, C., et al.: Deep people counting in extremely dense crowds. In: Proceedings ACM international Conference on Multimedia, pp. 1299–1302, (2015)

37. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of the Neural Information Processing, pp. 1097–1105, (2012)

38. Wang, C., et al.: Faster and Lighter: Real-time crowd counting with deep learning. In: Proceedings of the British Machine Vision Conference, (2012)

39. Zhang, C., et al.: Cross-scene crowd counting via deep convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3618–3626, (2018)
44. Liu, Y., et al.: Point in, box out: beyond counting persons in crowds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6469–6478, (2019)
45. Gao, J., Wang, Q., Li, X.: PCC net: perspective crowd counting via spatial convolutional network. IEEE Trans. Circuits Syst. Video Technol. 30(10), 3486–3498 (2019). https://doi.org/10.1109/TCSVT.2019.2919139
46. Liu, N., et al.: Adcrowdnet: an attention-injective deformable convolutional network for crowd understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3225–3234, (2019)
47. Zeng, X., et al.: Dspnet: deep scale purifier network for dense crowd counting. Expert Syst. Appl. 141, 112977 (2020)
48. Shi, Z., et al.: Crowd counting with deep negative correlation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5382–5390, (2018)
49. Huang, G., et al.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708, (2017)
50. Chen, X., et al.: Scale pyramid network for crowd counting. In: Proceedings of the IEEE Winter Conference on Applications of Computer, pp. 1941–1950, (2019)
51. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141, (2018)
52. Idrees, H., et al.: Composition loss for counting, density map estimation and localization in dense crowds. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss Y. (eds) Computer Vision – ECCV 2018. Lecture Notes in Computer Science, vol. 11206, pp. 532–546. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01216-8_33
53. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv:1412.6980 (2014)
54. Paszke, A., et al.: Pytorch: Tensors and dynamic neural networks in python with strong GPU acceleration (2017)
55. Sam, D.B., Babu, R.Y.: Top-down feedback for crowd counting convolutional neural network. In: AAAI Conference Artificial Intelligence, (2018)
56. Shi, Z., et al.: Multiscale multitask deep netvlad for crowd counting. IEEE Trans. Industrial Informatics. 14(11), 4953–4962 (2018)
57. Liu, X., van de Weijer, J., Bagdanov, A.D.: Leveraging unlabeled data for crowd counting by learning to rank. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7661–7669, (2018)
58. Liu, C., Weng, X., Mu, Y.: Recurrent attentive zooming for joint crowd counting and precise localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1217–1226, (2019)
59. Hossain, M., et al.: Crowd counting using scale-aware attention networks. In: Proceedings of the IEEE Winter Conference on Applications of Computer, pp. 1280–1288. Lake Tahoe (2019)
60. Jiang, X., et al.: “Crowd counting and density estimation by trellis encoder-decoder networks.” In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1217–1226, (2019)

How to cite this article: Wang L, Li Y, Peng S, Tang X, Yin B. Multi-level feature fusion network for crowd counting. IET Comput. Vis. 2021;15:60–72. https://doi.org/10.1049/cvi2.12012