Neurocomputing 409 (2020) 420–430

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Crowd counting via scale-communicative aggregation networks

Lixian Yuan [a], Zhilin Qiu [a], Lingbo Liu [a], Hefeng Wu [a,*], Tianshui Chen [b], Pei Chen [a], Liang Lin [a]

[a] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
[b] DarkMatter AI Research, Guangzhou, China

ARTICLE INFO

Article history:
Received 16 December 2019
Revised 28 March 2020
Accepted 15 May 2020
Available online 6 June 2020
Communicated by Wang QI

Keywords:
Crowd counting
Scale-communicative
Aggregation networks

ABSTRACT

Crowd counting is a fundamental computer vision task that draws increasing attention in recent years, due to its wide applications in commercial activities and public securities. Despite much process has been achieved by applying the resurgent neural networks in this task, critical challenges still lie in tremendous variation of crowd scales, together with other issues like background clutters and occlusions, making the crowd appearances hard to model. To address these challenges, we propose a scale-communicative aggregation network (SCANet) for crowd counting. Our model is characterized by three aspects: (i) It contains different streams of convolutional neural networks (CNNs), where each stream consumes an individual scaled version of the input image and communicates complementarily to produce a high-resolution density map. (ii) Each CNN stream obtains robust feature presentation via our proposed multi-scale feature encoders (MSFEs) with dilated convolutional layers, and skip connections are adopted to exploit multi-stage feature aggregation. (iii) A multi-scale structural similarity metric along with Euclidean distance is introduced for optimizing the quality of generated density maps. Extensive experiments and comparisons on several crowd counting benchmarks demonstrate the effectiveness of our proposed method.

1. Introduction

Crowd counting has become an increasingly important problem due to its wide applications in supporting daily economic and social demands. For example, an accurate estimate of the crowd is critical for public activities ranging from commercial statistics to crowd control, abnormal event detection, and other security tasks. Though counting the number of people in crowd images is a straightforward problem, it still remains challenging due to large variation of crowd scales and diversified crowd distributions.

As shown in Fig. 1, people vary from several pixels to a large region in scales and from single person to several hundreds in crowd densities. The current leading methods usually estimate the crowd number via generating a crowd density map from the scene image instead of directly regressing the total crowd number. The generated crowd density map is then used for adding up to obtain the crowd count in this task, and it can also be further exploited for other related tasks like crowd behavior analysis.

Recently, crowd counting methods built on Convolutional Neural Network (CNN) backbones have achieved impressive performance [1–5], due to the powerful representation learning ability of the deep CNN models [6–11]. These methods generally handle the scale variation problem by utilizing the multi-column architectures to enhance feature learning, where the input is processed with convolutional kernels of different sizes in each column so as to extract features on different scales. However, they suffer from certain issues. First, the large scale variation cannot be covered by the limited number of columns and blindingly increasing the column number will lead to massive parameter overload, which easily fails on diversified scales feature leaning, as revealed by [1]. Second, these networks usually generate low-resolution crowd density map while learning increasingly abstract features. A low-resolution density map is insufficient to tackle with tiny heads in crowd scenes and estimate the accurate crowd count. Third, the pixel-wise Euclidian loss between the predicted and ground-truth density maps easily leads the trained model to generate a criticized blurring density map. Therefore, they still have the problem of severe accuracy degradation when applied in challenging crowd scenes with large scale variation or high congestion.

In order to alleviate the influence of these drawbacks aforementioned, we propose a novel neural network framework, termed as Scale-Communicative Aggregation Network (SCANet), which comprehensively integrates multi-scale features and obtains high-resolution density maps for crowd counting. We introduce an effective Multi-Scale Feature Encoder (MSFE) to extract

* Corresponding author.
E-mail address: wuhefeng@gmail.com (H. Wu).

https://doi.org/10.1016/j.neucom.2020.05.042
0925-2312/

Fig. 1. Example crowd scenes and their ground-truth density maps. There exist large variations of crowd scales in different scenes, and generating the corresponding crowd density map remains challenging.

multi-scale robust feature representation. Each MSFE consists of multiple columns of stacked CNNs and exploits dilated convolutional layers to enlarge receptive fields. With different dilation rates, these columns have various receptive fields and can respectively model the appearance of people on different scales. Then more diversified scale features are aggregated through the serial connection of MSFEs. We further propose a scale communication architecture for consuming different scales of input image to obtain high-resolution density map. Each scale stream processes a different scaled version of input image and provides complemental information for each other. The whole network finally aggregates multi-stage and multi-scale features into a high-resolution representation, thereby generating a high-resolution density map for accurate crowd counting. Moreover, we exploit a multi-scale structural similarity loss to enforce our network to learn the local correlation of multi-scale patches from the density maps, which better capture the crowd density distribution than those learned from solely pixel-wise single-scale consistency loss. Our contributions can be summarized as follows:

- We propose a Scale-Communicative Aggregation Network (SCA-Net) to deal with the variations of people scales in complex scenes. We exploit stacked Multi-Scale Feature Encoders to extract robust multi-scale features and introduce the scale communication architecture between multi-scale inputs to generate the high-resolution density map.
- We incorporate a novel training loss, named Multi-Scale Structural Similarity (MS-SSIM) loss, along with the Euclidean Distance loss to force the network to learn the local correlations of patches on the density maps, which helps generate high-quality density map and obtain accurate crowd count.
- Extensive experiments on several challenging benchmarks show the remarkable performance of our proposed method in comparison with other state-of-the-art methods.

2. Related work

In this section, we review the recent related works on crowd counting in computer vision community. We categorize them into detection based methods and regression based methods.

Crowd counting based on detection. It is straightforward to analyze the crowding counting problem as a head or person detection problem as the main entities are persons in existing crowd counting datasets. Early researchers [12] apply typical detection schemes with sliding windows to first detect the persons in a single image and count their number afterwards. Hand-crafted features, such as Haar wavelets [13], HOG [14], or their combinations are extracted from a human body or particular human parts for detection. These methods usually hold the assumption that the crowd consists of separate human entities, which can be easily detected by trained human detectors. Such detection based methods are limited by occlusions. Though particular body parts are detected to addresss this problem [15], they still fail in dense crowds with complex clutter scenes.

Crowd counting based on regression. Due to the restriction of detection based methods on crowd counting, the regression based methods gain more attention and now have been the most extensively used methods. In [16], researchers use Bayesian Poission regression for crowd counting from low-level features. Following that, multiple sources, such as head detections, texture elements and Fourier analysis are used to regress crowd count in [17]. In [18], Lempitsky et al. casted the problem as estimating the counts of objects within an image region instead of a whole image. While the mapping between image patches features and relative locations of objects in corresponding patches are learnt in [19] to generate the density map.

Utilizing deep learning in computer vision has advanced many research problems, and the CNN-based crowd counting methods are investigated and obtain remarkable progress. A classic Alexnet style architecture in [20] is trained to regress crowd counts. In [21], layered boosting and selective sampling are incorporated during the learning process of the network. Shang et al. proposed an end-to-end CNN architecture in [22], which takes a whole image as input and directly regresses the counting results by taking advantages of contextual information when predicting both local and global counts. Boominathan et al. [23] proposed a dual-column architecture. The VGG deep neural network in the architecture is designated for sparse crowds while the shallow network to capture the dense crowds. Zhang et al. [24] proposed a CNN for crowd counting in different scenes. But this method is limited as it requires perspective maps both on training and test scenes, which is not available in practical applications. A multi-column convolutional neural structure is proposed in [1]. The network is layered with different receptive fields in different columns to

capture different scale features, hence capture crowds at different scales. Sam et al. [25] adopted the same multi-column architecture and performed a different training procedure to boost multi-scale feature learning. In [26], Shen et al. used a multi-scale U-net structured generation network to attenuate the blurry effect of generating the density map, while Li et al. proposed CSRNet [27] with dilated convolutional layers to boost performance on crowd counting.

3. Proposed framework

In this section, we first briefly present the formulation of the crowd counting problem and then introduce our proposed framework in detail.

3.1. Problem formulation

Given an input image I, the crowd counting task requires a model F to predict the number C_{count} of people that appear in the image, which can be formulated as follows:

$$C_{count} = F(I). \tag{1}$$

It is observed in previous research that, instead of directly regressing the number of people, density-based crowd counting methods achieve much better performance [20,27–30]. This kind of methods estimates the crowd density map D_I of the input image I, and then the predicted number of people is obtained by adding up each pixel value of the density map. The formulation turns into:

$$C_{count} = F_s(D_I),\ \ D_I = F_d(I), \tag{2}$$

where F_s denotes the summation function, and F_d is the model aimed at estimating the crowd density map. Then the essential focus lies in how to build an appropriate model F_d.

3.2. Framework Overview

To address the crowd counting task, we propose a novel Scale-Communicative Aggregation Network (SCANet) model for generating high-quality crowd density maps. The proposed SCANet framework is illustrated in Fig. 2. Our framework is composed of two convolutional neural networks (CNNs) streams, each of which consists of several Multi-Scale Feature Encoders (MSFE) and learns increasingly robust feature presentation. An MSFE is designed to obtain richer feature representation and conserve scale diversities. One stream of convolutional neural network takes the original crowd scene image as input, while the other stream takes an upsampled input. Communications between streams are conducted to serve as the complement in resolutions for the output of the final high-resolution density map. Moreover, skip connections are utilized to exploit multi-stage feature aggregation and facilitate better gradient back-propagation, which concatenate and fuse multi-stage feature outputs from MSFEs in the final generation of crowd density map.

3.3. Feature aggregation with multi-scale feature encoder

Inside each stream of convolutional neural network architecture, we propose a unified neural network module, named Multi-Scale Feature Encoder (MSFE), to tackle with scale variation of crowds in scenes. We develop our MSFE with multiple columns of CNNs as previous work [1]. Each column in MSFE is designed to handle the scale variation of crowds. In previous work [1], the multi-column CNNs are stacked with normal convolutional layers and use different kernel sizes and channels in each column. Differently, We adopt dilated convolution layers with various dilation rates in our proposed MSFE to enlarge the receptive fields during feature extraction. Dilated convolutional layers [31] serve as a good alternative of pooling layer, which use sparse kernels to enlarge receptive fields and extract deeper features without losing spatial resolutions or increasing parameters. A demonstration of dilated convolutions is shown in Fig. 3. In dilated convolution, a small-size kernel with $k \times k$ filter can be enlarged to $k + (k-1)(r-1)$ with a dilation rate r. When the dilation rate is set to 1, the dilated convolution is equal to a normal convolutional layer. Thus the dilated convolutional layers allow flexible aggregation of the multi-scale contextual information and keeps the same resolution as well. As shown in Fig. 3, a larger dilated ration in (b) and (c) deliver a 5×5 and 7×7 receptive fields respectively comparing to a normal convolution layer with a filter kernel size of 3×3 (the dilation rate is set to 1) in (a).

The detail of our proposed MSFE is illustrated in Fig. 4. Similar to MCNN [1], a good example of our MSFE consists of three rows of stacked CNNs, each of which has four dilated convolutional layers. In the first row, the dilation rates of the dilated convolutional layers are set to 1, which is the same as normal convolutions. The second and especially the third rows adopt higher dilation rates in the dilated convolutional layers to enlarge the receptive fields and extract features. The dilated convolutional layers at the same column have the same kernel size and channel number, and share parameters. A ReLU layer is followed after each dilated convolutional layer. Finally, we integrate the output features from three rows with an element-wise maximization operation as the scale robust representations.

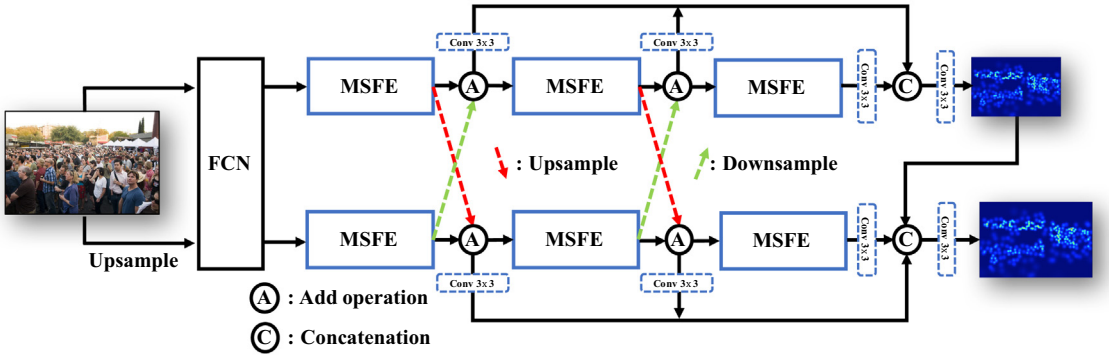

Fig. 2. Illustration of the proposed framework. Our framework is composed of dual streams of convolutional networks. The top network takes the original image as the input while the bottom network takes an upsampled image. We build the inter-communications among MSFEs from different networks to maintain the high-resolution output. Skip connections are also conducted for utilizing multi-stage feature outputs in generation of the final density map.

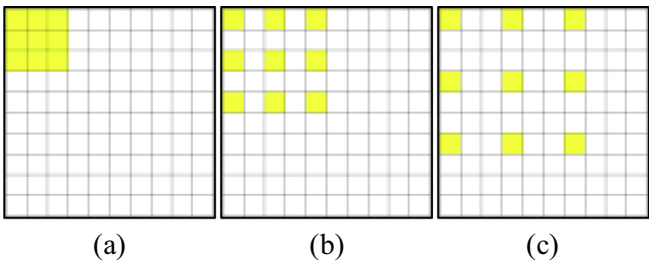

Fig. 3. Dilated convolutions with a kernel size of 3×3. (a) indicates that dilated convolution is the same as a normal convolutional layer when the dilation rate is set to 1. (b) and (c) have larger receptive fields respectively with the dilation rates setting to 2 and 3.

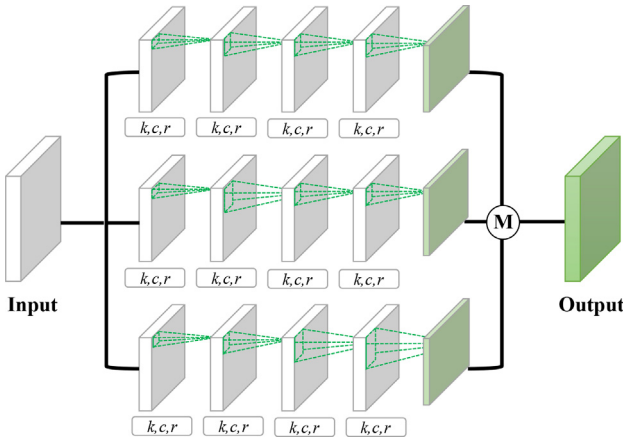

Fig. 4. Illustration of the proposed Multi-Scale Feature Encoder. This MSFE example is developed with 3 rows of stacked dilated convolutional networks, each of which has 4 dilated convolutional layers. (k, c, r) denotes kernel size, channels and dilation rates respectively. For the better capture of rich representations, the k, c and r will be set to different values at different rows. A ReLU layer is followed after each dilated convolutional layer. The output is integrated by an element-wise maximization operation.

3.4. Communicative CNN streaming networks for multi-scale inputs

The existing methods utilize the convolutional neural networks with multiple pooling layers to generate the density maps in low resolution. Although pooling layers are typically used for maintaining invariance and avoiding overfitting, they also reduce the spatial resolution thus are unable to localize the tiny heads. In our proposed methods, we conduct multi-scale stream fusions to obtain high-resolution representations to handle this issue. Specifically, we build our model upon two streaming CNNs. The top CNN stream takes the original image as the input to extract the feature and generate a coarse density map, which roughly localizes the crowd regions. The bottom CNN stream takes the image with two times upsampled size, and generate the final high-resolution density map with communication between the two CNN streams.

Our proposed framework architecture is composed of dual streams of convolutional neural networks, each of which consumes an individual scaled version of the input image. Both the top and bottom networks share a same front-end Fully Convolutional Network (FCN) and consist of three stacked Multi-Scale Feature Encoder (MSFE). The FCN is taken from the first ten layers of VGG16 [32] with three pooling layers. Given an image I of size $H \times W$, where H and W are the height and width of the image respectively, we feed the original image to the top network (scale 1) and upsampled image with $2H$ and $2W$ to the bottom network (scale 2). Thus the output size of the top network is $\frac{H}{s} \times \frac{W}{s}$ and the bottom network is $\frac{2H}{s} \times \frac{2W}{s}$ after the feature extraction in FCN, where s refers to the scale ratio depending on the pooling layers in FCN. The feature generated by the i-th MSFE of the top network is denoted as f_1^i and that of the bottom network as f_2^i. After the i-th MSFE in both networks, we conduct the a feature fusion by adding the feature outputs together from the both networks to communicate the complemental representation [33]. Specifically, f_1^i is upsampled as the same size as f_2^i, while f_2^i is downsampled via a convolutional layer with a stride of 2. By adding the features respectively, we have,

$$f_1^{i\prime} = f_1^i \oplus ds\left(f_2^i\right) \tag{3}$$

$$f_2^{i\prime} = f_2^i \oplus us\left(f_1^i\right) \tag{4}$$

where $\oplus$ is the element-wise add operation, ds is the downsample operation via convolution and us is a bilinear upsample operation. $f_1^{i\prime}$ and $f_2^{i\prime}$ are fed to the subsequent MSFE afterwards. We conduct a repeated feature fusion operation each time the networks pass an MSFE. The top network repeatedly received the complement feature information from the bottom network and vice versa. The communicative feature fusion operation is shown in Fig. 5.

Similar as Deeply-supervised nets proposed in [34], we regress a density map m^i via a convolutional layer with kernel size 3×3 each time when the network passes an MSFE and use it for later multi-stage feature fusion, which is called skip connections. By associating the local outputs to the final output, it serves as a feature regularization and helps accelerate the convergence with the deep supervision on side response. On the top network, we obtain a coarse density map $m_1 \in R^{\frac{H}{s} \times \frac{W}{s}}$ by feeding the concatenation of $m_1^1, m_1^2, \ldots, m_1^K$ into a weighted-fusion convolutional layer with kernel size 3×3, where K corresponds to the number of MSFEs in the network setting. We believe that the density map m_1 roughly localizes the crowd region, but it is not good enough to estimate the accurate number of people in the image due to its low-resolution representation. Therefore, we utilize the bottom network to generate a fine density map. Similar to the top network, the bottom network also outputs a side density map m_2^i each time when the network passes an MSFE. m_1 is upsampled to the same size of m_2^i. Thus $m_2^1, m_2^2, \ldots, m_2^K$ and the upsampled density map m_1 are concatenated to generate the final density map $m_2 \in R^{\frac{2H}{s} \times \frac{2W}{s}}$ via a 3×3 convolutional layer. m_2 exhibits more accurate spatial locations of the crowds and deals with the tiny heads. The whole network can be trained in an end-to-end manner.

3.5. Multi-scale structural similarity in local correlation learning

Most of the existing methods infer their models with the pixel-wise Euclidean distance, which leads to blurring density maps and inaccurate crowd count results. In [35], Cao et al. utilized a combination of Euclidean distance and single-scale structural similarity loss to train the network, but their estimated density maps are still far from satisfactory. In our framework, we propose a Multi-Scale Structural Similarity (MS-SSIM) loss to enforce our model to learn the local correlation of multi-scale patches on the density maps. We will first describe the single scale SSIM loss, and our MS-SSIM is modified based on single scale SSIM but is more effective to measure the similarity between generated density map and the ground-truth density map.

3.5.1. Euclidean distance measure

Euclidean distance usually serves as the simple yet insufficient measurement of difference between generated map and ground-truth density map. The loss is defined as follows:

$$L(\Theta) = \frac{1}{2N}\sum_{i=1}^{N}||X_i - Y_i||_2^2 \tag{5}$$

where N is the batch size during training, X_i is the predicted density map of the i-th image and Y_i denotes the corresponding ground-truth density map.

3.5.2. Single-scale SSIM

SSIM computes the similarity between two images from three local statistics as a common evaluation metric in image quality assessment, i.e. mean, variance and covariance. Following the setting in [36], a 5×5 normalized Gaussian kernel with a standard deviation of 1.5 is used to measure these three local statistics. The estimation can be easily implemented with a dilated convolutional layer. For each location $p = (i, j)$ on predicted map X, its local mean $\mu_X(p)$, variance $\sigma_X^2(p)$ and covariance $\sigma_{XY}^2(p)$ can be computed as:

$$\begin{aligned}
\mu_X(p) &= \sum_{\Delta i=-2}^{2}\sum_{\Delta j=-2}^{2} W(\Delta i, \Delta j)\cdot X\{p + (\Delta i, \Delta j)\cdot r\},\\
\sigma_X^2(p) &= \sum_{\Delta i=-2}^{2}\sum_{\Delta j=-2}^{2} W(\Delta i, \Delta j)\cdot \{X\{p + (\Delta i, \Delta j)\cdot r\} - \mu_X(p)\}^2,\\
\sigma_{XY}^2(p) &= \sum_{\Delta i=-2}^{2}\sum_{\Delta j=-2}^{2} W(\Delta i, \Delta j)\cdot \{X\{p + (\Delta i, \Delta j)\cdot r\} - \mu_X(p)\}\\
&\quad \cdot\{Y\{p + (\Delta i, \Delta j)\cdot r\} - \mu_Y(p)\}
\end{aligned} \tag{6}$$

where $(\Delta i, \Delta j)$ is the offset from the center and W denotes the parameters of the normalized Gaussian kernel. r is the dilation rate used to control the receptive field region. The mean $\mu_Y(p)$ and variance $\sigma_Y^2(p)$ of the GT map Y is also calculated with the same formulation above. Thus, we can compute the luminance comparison L, contrast comparison C and structure comparison S between the predicted map X and the GT map Y as follows:

$$\begin{aligned}
L(X, Y) &= \frac{2\mu_X\mu_Y + c_1}{\mu_X^2 + \mu_Y^2 + c_1}, C(X, Y) = \frac{2\sigma_x\sigma_Y + c_2}{\sigma_X^2\sigma_Y^2 + c_2},\\
S(X, Y) &= \frac{\sigma_{XY} + c_3}{\sigma_X\sigma_Y + c_3}
\end{aligned} \tag{7}$$

where c_1, c_2 and c_3 are small constants. The SSIM loss is defined as:

$$L_{\text{SSIM}} = 1 - L(X, Y)\cdot C(X, Y)\cdot S(X, Y) \tag{8}$$

3.5.3. Multi-Scale SSIM

For better capturing the crowd distribution than those solely pixel-wise or single scale consistency loss, we argue that density map should be estimated with its corresponding groundtruth in multi-scale local correlation. We build a CNN with M dilated convolutional layers, whose parameters are set to the fixed Gaussian kernel W as described earlier. Specifically, M is set to 5 in our proposed framework and the dilation rates are set to 1, 2, 3, 4, and 9 respectively. Different dilation rates in dilated convolutional layers are designed to calculate the SSIM of larger regions, which measures the local correlations comparisons on different scales. We calculate the contrast and structure difference after each dilated convolutional layer. The luminance difference is computed only once at the last layer as in [37]. The MS-SSIM loss $L_{\text{MS-SSIM}}$ is defined as follows:

$$\begin{aligned}
L_{\text{MS-SSIM}} &= 1 - \left[L_{M-1}(X_{M-1}, Y_{M-1})^{\alpha_{M-1}}\right]\cdot\prod_{j=0}^{M-1}\left[C_j(X_j, Y_j)^{\beta_j}\right]\\
&\quad\cdot\left[S_j(X_j, Y_j)^{\gamma_j}\right]
\end{aligned} \tag{9}$$

where α_j, β_j and γ_j are used to adjust the relative importance of different comparisons and they are set as the same values in [37]. In our framework, both the top and bottom streams of CNNs are optimized with the MS-SSIM loss. The MS-SSIM loss operation is illustrated in Fig. 6.

3.5.4. Loss function

By integrating our proposed MS-SSIM loss and Euclidean distance loss, we have,

$$Loss = \phi\cdot L_{\text{MS-SSIM}} + \varphi\cdot L_{Euclidean} \tag{10}$$

where ϕ and φ are used to adjust the importance of MS-SSIM loss and Euclidean loss respectively.

Algorithm 1 shows the training procedure of our proposed method, which is implemented in an end-to-end manner.

Algorithm 1: Training procedure of the proposed SCANet model

Input: original input images $\{I_i\}_{i=1}^n$; groundtruth head annotations $\{A_i\}_{i=1}^n$
Output: the SCANet model F_d
1: generate density map groundtruths $\{G_i\}_{i=1}^n$ from $\{A_i\}_{i=1}^n$;
2: initialize network weights W;
3: **repeat**
4: upsample the images $\{I_i\}_{i=1}^n$ to obtain $2\times$ upsampled images $\left\{\widehat{I}_i\right\}_{i=1}^n$;
5: feed the input and upsampled images to the network model;
6: extract features of images through network streams while conducting communications between streams as in Eq. (3) and (4);
7: predict density maps $\{O_i\}_{i=1}^n$;
8: compute the loss L between density maps $\{O_i\}_{i=1}^n$ and groundtruths $\{G_i\}_{i=1}^n$ as in Eq. (10);
9: compute gradients ∇L;
10: update network weights $W \leftarrow W'$;
11: **until** program reaches maximum epochs;

4. Experiments

In this section, we first present our implementation details and experimental settings. Following, we report the experiment results of our proposed methods comparing with other state-of-the-art methods and evaluate the effectiveness of each component in our proposed framework.

4.1. Implementation details

4.1.1. Network implementations

Our proposed Scale-Communicative Aggregation Network is implemented with Pytorch toolbox. The Feature Convolutional Network (FCN) is taken from the first 10 layers of VGG model [32] with 3 pooling layers, and initialized with the pretrained model on ImageNet. The learning rate in our method is set to 1×10^{-5}. The filter wights in other convolutional layers are randomly initialized by Gaussian distributions with zero mean and standard deviation of 0.01. 16 crops of size 224×224 are sampled from the original image at each iteration. We train our model with an end-to-end manner for around 1000 epochs with Adam optimizer.

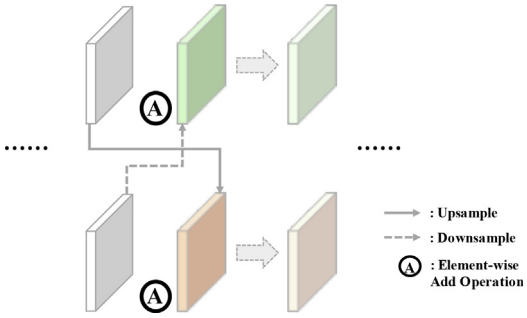

Fig. 5. Communicative feature fusion operation between two networks. The feature output from the previous MSFE is upsampled or downsampled to the same size of the other network and fused by an element-wise add operations.

4.1.2. Groudtruth generation

The groudtruth crowd density maps are generated with geometry-adaptive kernels as proposed in previous work [1]. Give the head annotations of a scene image, the distances to its n nearest neighbors are denoted as $\{d_1, d_2, \ldots, d_n\}$. We label this head via a normalized Gaussian kernel with spread $\sigma = s \cdot \frac{1}{N}\sum_{i=1}^{N} d_i$, where N is set to 3 and s is set to 0.3. The radius of the Gaussian kernel is $6\sigma \times 6\sigma$.

4.2. Evaluation criterion

The accuracy of the crowd counting estimation is usually evaluated via the Mean Absolute Error (MAE) and Mean Squared Error (MSE). The MAE and MSE are defined as follows,

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{C}_i - C_i| \tag{11}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|\hat{C}_i - C_i|^2} \tag{12}$$

where N denotes the number of testing samples, $\hat{C}_i$ is the predicted crowd count number as the sum of all pixel values on the generated density map. While C_i is the ground-truth crowd count number.

4.3. Evaluation benchmarks

We evaluate and compare our proposed method on three public crowd counting benchmarks, i.e., ShanghaiTech dataset [1], UCF_CC_50 [17] and UCF-QNRF [38].

4.3.1. ShanghaiTech dataset

ShanghaiTech dataset [1] is one of the largest datasets in terms of the number of annotated people scales. The dataset consists of two parts: Part_A and Part_B. There are 1198 annotated images with a total of 330,165 crowd number. Part_A contains 482 images and 300 of them are used for training and the remaining 182 images for testing. While Part_B has 716 images taken from the streets of Shanghai metropolitan areas. The training set of Part_B has 400 images, and the testing set has 316 images. The average crowd annotations in Part_A and Part_B are about 500 and 120 respectively.

4.3.2. UCF-QNRF dataset

UCF-QNRF [38] is one of the most challenging datasets due to its diversified viewpoints, densities and various crowd scales. There are 1,535 images with 1,251,642 annotations in UCF-QNRF datasets. The training set and testing set are split as 1201 and 334 images respectively. In this dataset, the minimum and maximum crowd counts are 49 and 12,865, while the median and mean crowd counts are 425 and 815, respectively.

4.3.3. UCF_CC_50 dataset

The UCF_CC_50 dataset [17] only has 50 crowd images collected from the Internet. The crowd numbers of the dataset vary from 94 to 4543 with an average of 1280 persons. The limited number of images and the large range in crowd counts among images make this dataset also a challenging one.

4.4. Comparison results

We report our results on these benchmarks in Table 1, compared with 12 existing leading methods, including MCNN [1], CMTL [39], Switch-CNN [25], CP-CNN [2], PCCNet [40], CSRNet [27], SANet [35], SCAR [5], SFCN [41], ADCrowdNet [28], HA-CCN [42] and TEDnet [43]. In ShanghaiTech part_A, we obtain the lowest MAE and MSE among the compared methods. Especially, we achieve 2.5 and 1.2 lower in MAE and MSE than those of the existing best method HA-CCN [42]. Notice that some methods achieve low MAE results while the MSE results are relatively higher, such as TEDnet [43], while our proposed method can obtain low errors in both MAE and MSE. In ShanghaiTech Part_B, our proposed results are comparable and close to the best existing method in MAE while the MSE report is the lowest. Our outstanding results on ShanghaiTech dataset indicates that the proposed method is robust and effective in both congested scenes (ShanghaiTech part_A) and sparse scenes (ShanghaiTech part_B). Likewise, on the UCF_CC_50 dataset, our proposed method reports 248.7 and 334.5 in MAE and MSE, respectively. The results are better than

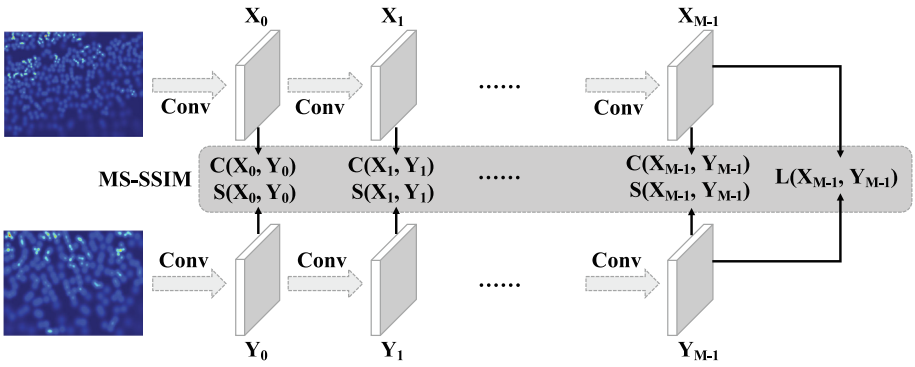

Fig. 6. Multi-Scale Structural Similarity loss. The contrast and structure comparisons are calculated after each dilated convolution, and the luminance comparison is only computed at the last later.

Table 1
Evaluation results on three standard benchmarks.

Methods	ShanghaiTech Part_A		ShanghaiTech Part_B		UCF_CC_50		UCF_QNRF	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MCMM [1]	110.2	173.2	26.4	41.3	377.6	509.1	277	426
CMTL [39]	101.3	152.4	20.0	31.1	322.8	397.9	252	426
Switch-CNN [25]	90.4	135.0	21.6	33.4	318.1	439.2	228	445
CP-CNN [2]	73.6	106.4	20.1	30.1	298.8	320.9	–	–
PCCNet [40]	73.5	124.0	11.0	19.0	240.0	315.5	148.7	247.3
CSRNet [27]	68.2	115.0	10.6	16.0	266.1	397.9	–	–
SANet [35]	67.0	104.5	8.4	13.6	258.4	334.9	–	–
SCAR [5]	66.3	114.1	9.5	15.2	259.0	374.0	–	–
SFCN [41]	64.8	107.5	7.6	13.0	214.2	318.2	102.0	171.4
ADCrowdNet [28]	63.2	98.9	7.7	12.9	257.1	363.5	–	–
HA-CCN [42]	62.9	94.9	8.1	13.4	256.2	348.4	118.1	180.4
TEDnet [43]	64.2	109.1	8.2	12.8	249.4	354.5	113	188
Ours	**60.4**	**93.7**	**8.5**	**12.5**	**248.7**	**334.5**	**104.3**	**179.8**

those of other compared methods. For the UCF_QNRF dataset, we rank top two of the lowest MAE and MSE metrics comparing with state-of-the-art methods. We report very close results in both MAE and MSE to the best-performing method SFCN [41]. One reason that our proposed method performs slightly worse than SFCN is that we resize the input images from UCF_QNRF in preprocessing, which may lose some important information in the original input images. Example results of generated density maps are shown in Fig. 7. It can be observed that in both sparse and congested crowd scenes, our method managed to narrow the gap between crowd count prediction and ground-truth by generating good high-resolution density maps. To summarize, our proposed method effectively handles various scales of crowds in crowd counting and achieves comparable, even the best performance to the state-of-the-art methods.

4.5. Ablation study

In this section, we report our ablation study results on ShanghaiTech Part_A dataset to evaluate the contribution of each component in our proposed framework for crowd counting.

4.5.1. Multi-scale feature encoders

We evaluate the crowd estimate results by comparing the networks with and without our proposed Multi-Scale Feature Encoders. Three network variants are evaluated in our ablation studies, i.e., our framework leaving out all the MSFEs, with one-row MSFEs, and three-row MSFEs. As shown in Table 2, the performance of leaving out the MSFEs, which directly regresses the feature outputs from FCN, performs poorest among the three network variants. And the MSFE with one-row network performs poorer than the three-row MSFEs. Our proposed framework with three-row MSFEs reaches the highest performance, indicating that enlarging the receptive fields and integrating rich multi-scale feature representations serve as an effective manner in crowd counting.

We also conduct experiments to show how the number of MSFEs affects the final performance of our proposed framework. Table 3 reports experiment results. From Table 3, we can see that our method employing three MSFEs in the network structure outperforms that of using two or four MSFEs. Fewer MSFEs are not able to generate rich features for accurate crowd estimate while blindly adding MSFEs in the network might cause loss to feature

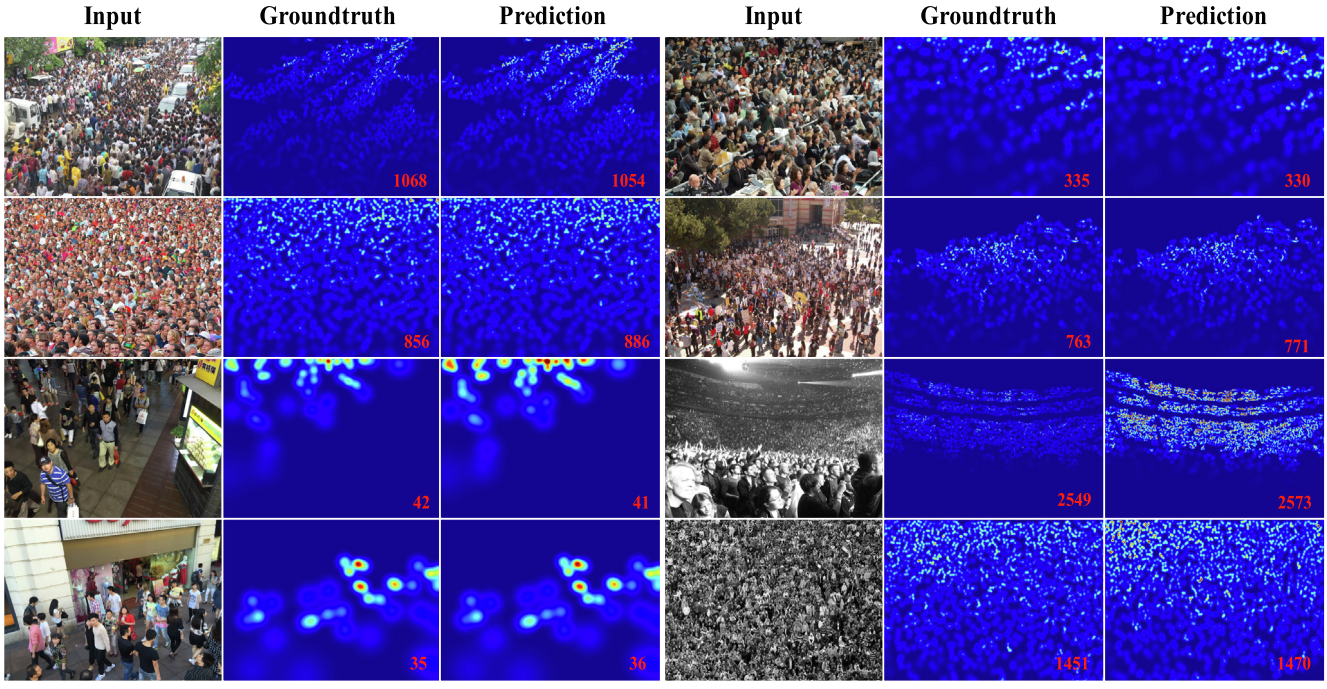

Fig. 7. From the left to right are the original input images, groundtruth density maps and our generated density maps. Our proposed methods can generate high quality density maps and predict a close crowd count to the groundtruth count both in sparse and dense crowd scenes. Zoom in for better observations.

Table 2
Estimated predictions with different configurations of MSFE or without MSFE.

Configurations of MSFEs	MAE	MSE
W/O MSFE	66.2	107.0
W/ One-Row MSFE	64.5	100.1
W/ Three-Row MSFE	60.4	93.7

representation of the original image details with much heavier computational cost.

Apart from the influences of the structure and number of our proposed MSFEs, we explore the fusion strategy of the output from the last layers of MSFEs. We conduct comparison experiments on three types of strategies, i.e., element-wise maximization, which is adopted in our method, concatenation and 1×1 convolution. Table 4 shows the comparison results. From Table 4, we see that element-wise maximization outperforms other fusion strategies in our proposed method. The reason may be that concatenation takes all information from the output without selection, which makes the feature more obscure. 1×1 convolution works better yet still causes information smoothing to the features, while element-wise maximization preserves the highlight in the features, providing critical and discriminative features for generating the final density maps.

4.5.2. Inter-communications between local outputs

We also conduct the experiments with and without the inter-communications between local outputs. The inter-communication plays as a complement role for both low and high resolution representations. The results are shown in Table 5. By building the inter-communications between local outputs, the final generated density map can better captures detailed information and result in a more accurate crowd count prediction.

4.5.3. The resolution of the output density maps

The output density map is used to add up for the final crowd count. We conduct experiments to show how the resolution of the density maps affects the performance in crowd counting scenarios. We first utilize one stream of our proposed network for the generation of low-resolution density maps. Then we utilize the full network structure of our proposed method for generation of high-resolution density maps to see the performance improvement on account of the complementary information from two times upsampled images. Table 6 shows our experiment results. From Table 6, we can see that fine high-resolution density maps obtain more accurate crowd count than coarse low-resolution density maps. However, blindly upsampling input images more to obtain higher resolution of density maps is not practical. First, it will incur much more uncertainties that can mislead the model. For example, upsampling the original image by 4 times will need to generate 15 times more pixels than the original image and may provide little information gain along with great uncertainty. Second, the computational expense increases rapidly. When our SCANet model takes the $2\times$ upsampled image, it requires about 8 GB GPU memory and can be handled with a GeForce GTX 1080TI GPU that has about 11 GB memory. With a $4\times$ upsampled image, the GPU memory requirement is more than 27 GB. Therefore, a $2\times$ upsampled image is more practical.

Table 3
Estimated predictions with different numbers of MSFEs.

Numbers of MSFEs	MAE	MSE
0 MSFE	66.2	107.0
1 MSFE	64.6	99.6
2 MSFEs	64.0	103.1
3 MSFEs	60.4	93.7
4 MSFEs	63.5	97.8

Table 4
The influences of different fusion techniques to the output of the layers in MSFEs.

Fusion Strategy in MSFEs	MAE	MSE
Concatenation	65.1	98.7
1×1 Convolution	63.3	97.6
Element-wise maximization	60.4	93.7

Table 5
Evaluation results with and without inter-communications between CNN streams

Networks	MAE	MSE
W/O Inter-Communications	62.5	98.3
W/ Inter-Communications	60.4	93.7

Table 6
The impact of density map resolution on final crowd count estimation.

Resolutions of density maps	MAE	MSE
low-resolution (coarse)	61.8	94.4
high-resolution (fine)	60.4	93.7

4.5.4. Multi-Scale Structural Similarity loss measurement

In the criteria loss part, we compare our proposed loss with the Euclidean criterion only, SSIM only, and their combinations. We also compare networks trained under only MS-SSIM without Euclidean loss. The results in Table 7 show that a combination of our proposed MS-SSIM and Euclidean loss reaches the best performance as it also learns the local correlations of patches in generated density maps.

In Fig. 8, we show some density map comparison results of the proposed framework with different network architectures on ShanghaiTech part_A. As can be seen in Fig. 8, the first two columns are the input images and their corresponding groudtruth density maps. The third column refers to the proposed framework with one-row MSFE, while the forth column refers to three-row MSFE. And the last column is the results with inter communications between dual networks in our proposed framework. The results demonstrate that three-row MSFE outperforms one-row MSFE and obtain richer feature representations, and the inter-communications between dual networks in the proposed framework generates high resolution density maps and closer crowd counts to the groundtruths.

4.5.5. Feature extraction backbone networks

We also evaluate different feature extraction backbone networks to obtain the FCN feature outputs for the proposed framework. As shown in Table 8, we tried ResNet 34, ResNet 50, ResNet 101 and VGG 16. The results show that the VGG 16 outperforms other feature extraction backbone networks.

Table 7
Evaluation results under different similarity measurements

Scales	Criterion	MAE	MSE
Single	Euclidean Distance	68.9	112.7
	SSIM	79.8	140.3
	SSIM + Eucli Dis	68.3	109.8
Multi	MSSSIM	61.8	96.6
	MSSSIM + Eucli Dis	60.4	93.7

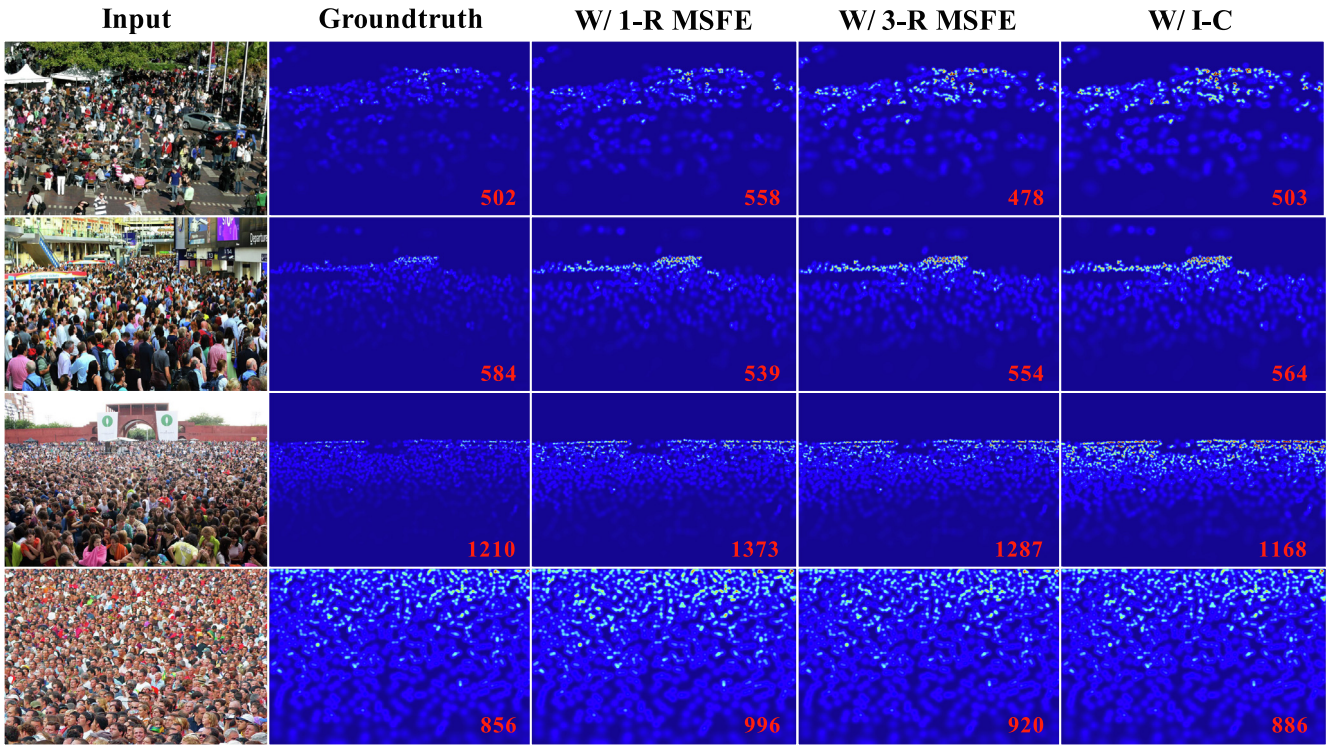

Fig. 8. From the left to right are the original input images from ShanghaiTech part_A, groundtruth density maps and three density maps generated by our proposed framework with different architectures, with one-row proposed MSFE, with three-row MSFE and with inter-communications. As we can see, the density maps are of higher resolutions and the corresponding crowd counts get closer to the groundtruths, which proves that our proposed modules in our framework are effective in solving the crowd counting problem.

Table 8
Evaluation results with different backbone networks in our proposed framework

Backbone	MAE	MSE
ResNet-34	69.5	106.0
ResNet-50	66.6	102.0
ResNet-101	65.6	101.2
VGG-16	60.4	93.7

5. Conclusion

In this paper, we proposed a Scale-Communicative Aggregation network for crowd counting. We develop a Multi-Scale Feature Encoder utilizing dilated convolutional layers with multiple dilation rate to extract richer feature representations.Inter-communications between local outputs from dual networks are conducted to maintain the high-resolution representations. We also integrate the Multi-Scale Structural Similarity loss with Euclidean loss to measure the difference between generated density maps and their corresponding groundtruths. Experiments on several standard benchmarks show that our proposed method reach comparable results with other state-of-the-art methods in crowd counting.

CRediT authorship contribution statement

Lixian Yuan: Conceptualization, Methodology, Software, Writing - original draft. **Zhilin Qiu:** Data curation, Methodology, Software, Investigation. **Lingbo Liu:** Software, Investigation, Writing - review & editing. **Hefeng Wu:** Conceptualization, Methodology, Writing - review & editing. **Tianshui Chen:** Visualization, Validation. **Pei Chen:** Supervision, Writing - review & editing. **Liang Lin:** Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was partly supported by the National Natural Science Foundation of China (61876045), Zhujiang Science and Technology New Star Project of Guangzhou (201906010057), and Natural Science Foundation of Guangdong Province (2017A030312006).

References

[1] Y. Zhang, D. Zhou, S. Chen, S. Gao, Y. Ma, Single-image crowd counting via multi-column convolutional neural network, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 27-30, 2016, pp. 589–597.
[2] V.A. Sindagi, V.M. Patel, Generating high-quality crowd density maps using contextual pyramid cnns, in: Proceedings of IEEE International Conference on Computer Vision (ICCV), Venice, Italy, October 22-29, 2017, pp. 1879–1888.
[3] L. Liu, H. Wang, G. Li, W. Ouyang, L. Lin, Crowd counting using deep recurrent spatial-aware network, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, July 13-19, 2018, pp. 849–855.
[4] J. Ma, Y. Dai, Y. Tan, Atrous convolutions spatial pyramid network for crowd counting and density estimation, Neurocomputing 350 (2019) 91–101.
[5] J. Gao, Q. Wang, Y. Yuan, SCAR: spatial-/channel-wise attention regression networks for crowd counting, Neurocomputing 363 (2019) 1–8.
[6] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 27-30, 2016, pp. 770–778.
[7] T. Chen, L. Lin, L. Liu, X. Luo, X. Li, DISC: deep image saliency computing via progressive representation learning, IEEE Trans. Neural Netw. Learn. Syst. 27 (6) (2016) 1135–1149.
[8] H. Li, H. Wu, H. Zhang, S. Lin, X. Luo, R. Wang, Distortion-aware correlation tracking, IEEE Trans. Image Process. 26 (11) (2017) 5421–5434.

[9] T. Chen, M. Xu, X. Hui, H. Wu, L. Lin, Learning semantic-specific graph representation for multi-label image recognition, in: Proceedings of IEEE International Conference on Computer Vision (ICCV), Seoul, Korea (South), October 27–November 2, 2019, pp. 522–531.
[10] H. Wu, Y. Hu, K. Wang, H. Li, L. Nie, H. Cheng, Instance-aware representation learning and association for online multi-person tracking, Pattern Recognition 94 (2019) 25–34.
[11] R. Chen, T. Chen, X. Hui, H. Wu, G. Li, L. Lin, Knowledge graph transfer network for few-shot recognition, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI), 2020.
[12] P. Dollár, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: an evaluation of the state of the art, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2012) 743–761.
[13] P.A. Viola, M.J. Jones, Robust real-time face detection, Int. J. Computer Vision 57 (2) (2004) 137–154.
[14] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–26 June, 2005, pp. 886–893.
[15] P.A. Viola, M.J. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, Int. J. Computer Vision 63 (2) (2005) 153–161.
[16] A.B. Chan, N. Vasconcelos, Bayesian poisson regression for crowd counting, in: Proceedings of IEEE 12th International Conference on Computer Vision (ICCV), Kyoto, Japan, September 27–October 4, 2009, pp. 545–551.
[17] H. Idrees, I. Saleemi, C. Seibert, M. Shah, Multi-source multi-scale counting in extremely dense crowd images, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, June 23-28, 2013, pp. 2547–2554.
[18] V.S. Lempitsky, A. Zisserman, Learning to count objects in images, in: Proceedings of 24th Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, 2010, pp. 1324–1332.
[19] V. Pham, T. Kozakaya, O. Yamaguchi, R. Okada, COUNT forest: Co-voting uncertain number of targets using random forest for crowd density estimation, in: Proceedings of IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, December 7-13, 2015, pp. 3253–3261.
[20] C. Wang, H. Zhang, L. Yang, S. Liu, X. Cao, Deep people counting in extremely dense crowds, in: Proceedings of the 23rd Annual ACM Conference on Multimedia, Brisbane, Australia, October 26–30, 2015, pp. 1299–1302.
[21] E. Walach, L. Wolf, Learning to count with CNN boosting, in: Proceedings of 14th European Conference of Computer Vision (ECCV), Amsterdam, The Netherlands, October 11–14, 2016, pp. 660–676.
[22] C. Shang, H. Ai, B. Bai, End-to-end crowd counting via joint learning local and global count, in: Proceedings of IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, September 25–28, 2016, pp. 1215–1219.
[23] L. Boominathan, S.S.S. Kruthiventi, R.V. Babu, Crowdnet: A deep convolutional network for dense crowd counting, in: Proceedings of ACM Conference on Multimedia, Amsterdam, The Netherlands, October 15–19, 2016, pp. 640–644.
[24] C. Zhang, H. Li, X. Wang, X. Yang, Cross-scene crowd counting via deep convolutional neural networks, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, June 7–12, 2015, pp. 833–841.
[25] D.B. Sam, S. Surya, R.V. Babu, Switching convolutional neural network for crowd counting, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, July 21–26, 2017, pp. 4031–4039.
[26] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, X. Yang, Crowd counting via adversarial cross-scale consistency pursuit, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, June 18–22, 2018, pp. 5245–5254.
[27] Y. Li, X. Zhang, D. Chen, Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, June 18–22, 2018, pp. 1091–1100.
[28] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, H. Wu, Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, June 16–20, 2019, pp. 3225–3234.
[29] L. Liu, Z. Qiu, G. Li, S. Liu, W. Ouyang, L. Lin, Crowd counting with deep structured scale integration network, in: Proceedings of IEEE International Conference on Computer Vision (ICCV), Seoul, Korea (South), October 27–November 2, 2019, pp. 1774–1783.
[30] L. Liu, J. Chen, H. Wu, T. Chen, G. Li, L. Lin, Efficient crowd counting via structured knowledge transfer, ArXiv (2020) abs/2003.10120.
[31] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: Proceedings of 4th International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, May 2–4, 2016.
[32] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, May 7–9, 2015.
[33] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, (CVPR), Long Beach, CA, USA, June 16–20, 2019, pp. 5693–5703.
[34] C. Lee, S. Xie, P.W. Gallagher, Z. Zhang, Z. Tu, Deeply-supervised nets, in: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, San Diego, California, USA, May 9–12, 2015.
[35] X. Cao, Z. Wang, Y. Zhao, F. Su, Scale aggregation network for accurate and efficient crowd counting, in: Proceedings of 15th European Conference of Computer Vision (ECCV), Munich, Germany, September 8–14, 2018, pp. 757–773.
[36] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[37] Z. Wang, E.P. Simoncelli, A.C. Bovik, Multiscale structural similarity for image quality assessment, in: Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers, 2003, pp. 1398–1402.
[38] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Máadeed, N.M. Rajpoot, M. Shah, Composition loss for counting, density map estimation and localization in dense crowds, in: Proceedings of 15th European Conference of Computer Vision (ECCV), Munich, Germany, September 8–14, 2018, pp. 544–559.
[39] V.A. Sindagi, V.M. Patel, CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting, in: Proceedings of 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, August 29-September 1, 2017, pp. 1–6.
[40] J. Gao, Q. Wang, X. Li, PCC net: Perspective crowd counting via spatial convolutional network, ArXiv (2019) abs/1905.10085.
[41] Q. Wang, J. Gao, W. Lin, Y. Yuan, Learning from synthetic data for crowd counting in the wild, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, June 16–20, 2019, pp. 8198–8207.
[42] V.A. Sindagi, V.M. Patel, HA-CCN: hierarchical attention-based crowd counting network, IEEE Trans. Image Process. 29 (2020) 323–335.
[43] X. Jiang, Z. Xiao, B. Zhang, X. Zhen, X. Cao, D.S. Doermann, L. Shao, Crowd counting and density estimation by trellis encoder-decoder networks, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, June 16–20, 2019, pp. 6133–6142.

Lixian Yuan received the B.E. degree from the School of Software Engineering, South China University of Technology, in 2014. He is currently pursuing the Ph.D. degree in computer science and technology with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His current research interests mainly focus on computer vision and machine learning.

Zhilin Qiu received the B.E. degree from the School of Software, Sun Yat-sen University, Guangzhou, China, in 2016, where he received the Master's degree in computer science with the School of Data and Computer Science, in 2019. He currently works with ByteDance Technology Co., Ltd. His current research interests include computer vision, intelligent transportation systems, and parallel computation.

Lingbo Liu received the B.E. degree from the School of Software, Sun Yat-sen University, Guangzhou, China, in 2015, where he is currently pursuing the Ph.D. degree in computer science with the School of Data and Computer Science. From March 2018 to May 2019, he was a research assistant at the University of Sydney, Australia. His current research interests include machine learning and intelligent transportation systems. He has authorized and co-authorized on more than 10 papers in top-tier academic journals and conferences.

Hefeng Wu received the B.S. degree in Computer Science and Technology and the Ph.D. degree in Computer Application Technology from Sun Yat-sen University, China, in 2008 and 2013, respectively. He is currently a full Research Scientist with the School of Data and Computer Science, Sun Yat-sen University, China. His current research interests include computer vision, multimedia, and machine learning.

Tianshui Chen received the Ph.D. degree in computer science at the School of Data and Computer Science Sun Yat-sen University, Guangzhou, China, in 2018. Before that, he received the B.E. degree from the School of Information and Science Technology. He is currently a principal researcher at DMAI Co., Ltd. His current research interests include computer vision and machine learning. He has authored and coauthored approximately 20 papers published in top-tier academic journals and conferences. He has served as a reviewer for numerous academic journals and conferences, including TPAMI, TIP, TMM, TNNLS, CVPR, ICCV, ECCV, AAAI and IJCAI. He was the recipient of the Best Paper Diamond Award at IEEE ICME 2017.

Pei Chen received two Ph.D. degrees in wavelets and computer vision from Shanghai Jiaotong University, China, and Monash University, Melbourne, Australia, in 2001 and 2004, respectively. He was a postdoctoral researcher with Monash University, a senior research engineer with Motorola Labs, Shanghai, and a research professor with SIAT/CAS, Shenzhen, China. He is currently a full professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His current research interests include topics in computer vision and machine learning.

Liang Lin is a full professor at Sun Yat-sen University. From 2008 to 2010, he was a postdoctoral fellow at the University of California, Los Angeles. He led the SenseTime R&D teams to develop cutting-edge and deliverable solutions for computer vision, data analysis and mining, and intelligent robotic systems from 2016–2018. He has authored and coauthored more than 100 papers in top-tier academic journals and conferences (e.g., 15 papers in TPAMI and IJCV and 60 + papers in CVPR, ICCV, NIPS, and IJCAI). He has served as an associate editor of IEEE Trans. Human–Machine Systems, The Visual Computer, and Neurocomputing and as Area/Session Chair for numerous conferences, such as CVPR, ICME, ACCV, and ICMR. He was the recipient of the Annual Best Paper Award by Pattern Recognition (Elsevier) in 2018, Best Paper Diamond Award at IEEE ICME 2017, Best Paper Runner-Up Award at ACM NPAR 2010, Google Faculty Award in 2012, Best Student Paper Award at IEEE ICME 2014, and Hong Kong Scholars Award in 2014. He is a Fellow of IET.