

Crowd Counting with Deep Structured Scale Integration Network

Lingbo Liu¹, Zhilin Qiu¹, Guanbin Li¹, Shufan Liu³, Wanli Ouyang³ and Liang Lin^{1,2*}

¹Sun Yat-sen University ²DarkMatter AI Research

³The University of Sydney, SenseTime Computer Vision Research Group, Australia

Abstract

Automatic estimation of the number of people in unconstrained crowded scenes is a challenging task and one major difficulty stems from the huge scale variation of people. In this paper, we propose a novel Deep Structured Scale Integration Network (DSSINet) for crowd counting, which addresses the scale variation of people by using structured feature representation learning and hierarchically structured loss function optimization. Unlike conventional methods which directly fuse multiple features with weighted average or concatenation, we first introduce a Structured Feature Enhancement Module based on conditional random fields (CRFs) to refine multiscale features mutually with a message passing mechanism. In this module, each scale-specific feature is considered as a continuous random variable and passes complementary information to refine the features at other scales. Second, we utilize a Dilated Multiscale Structural Similarity loss to enforce our DSSINet to learn the local correlation of people's scales within regions of various size, thus yielding high-quality density maps. Extensive experiments on four challenging benchmarks well demonstrate the effectiveness of our method. Specifically, our DSSINet achieves improvements of 9.5% error reduction on ShanghaiTech dataset and 24.9% on UCF-QNRF dataset against the state-of-the-art methods.

1. Introduction

Crowd counting, which aims to automatically generate crowd density maps and estimate the number of people in unconstrained scenes, is a crucial research topic in computer vision. Recently, it has received increasing interests in both academic and industrial communities, due to its wide range of practical applications, such as video surveillance

*Corresponding author is Liang Lin. This work was supported in part by the National Key Research and Development Program of China under Grant No.2018YFC0830103, in part by the National Natural Science Foundation of China under Grant No.U1811463 and No.61976250, in part by National High Level Talents Special Support Plan (Ten Thousand Talents Program). This work was also sponsored by SenseTime Research Fund.



Figure 1. Visualization of people with various scales in unconstrained crowded scenes. The huge scale variation of people is a major challenge limiting the performance of crowd counting.

[40], traffic management [46] and traffic forecast [21].

Although numerous models [8, 11] have been proposed, crowd counting remains a very challenging problem and one major difficulty originates from the huge scale variation of people. As shown in Fig. 1, the scales of people vary greatly in different crowded scenes and capturing such huge scale variation is non-trivial. Recently, deep neural networks have been widely used in crowd counting and have made substantial progress [43, 25, 42, 24, 45, 22, 27]. To address scale variation, most previous works utilized multi-column CNN [47] or stacked multi-branch blocks [4] to extract multiple features with different receptive fields, and then fused them with weighted average or concatenation. However, the scale variation of people in diverse scenarios is still far from being fully solved.

In order to address scale variation, our motivations are two-fold. **First**, features of different scales contain different information and are highly complementary. For instance, the features at deeper layers encode high-level semantic information, while the features at shallower layers contain more low-level appearance details. Some related researches [41, 44] have shown that these complementary features can be mutually refined and thereby become more robust to scale variation. However, most existing methods use simple strategies (e.g., weighted average and concatenation) to fuse multiple features, and can not well capture the complementary information. Therefore, it is very necessary to propose

a more effective mechanism for the task of crowd counting, to fully exploit the complementarity between different scale features and improve their robustness.

Second, crowd density maps contain rich information of the people’s scales¹, which can be captured by an effective loss function. As shown in Fig. 2, people in a local region usually have similar scales and the radiuses of their heads are relatively uniform on density maps, which refers to the local correlation of people’s scales in our paper. Moreover, this pattern may vary in different locations. **i)** In the area near the camera, the radiuses of people’s heads are large and their density values are consistently low, thus it’s better to capture the local correlation of this case in a large region (See the green box in Fig. 2). **ii)** In the place far away from the camera, the heads’ radiuses are relatively small and the density map is sharp, thus we could capture the local correlation of this case in a small region (See the red box in Fig. 2). However, the commonly used pixel-wise Euclidean loss fails to adapt to these diverse patterns. Therefore, it is desirable to design a structured loss function to model the local correlation within regions of different sizes.

In this paper, we propose a novel Deep Structured Scale Integration Network (DSSINet) for high-quality crowd density maps generation, which addresses the scale variation of people from two aspects, including structured feature representation learning and hierarchically structured loss function optimization. First, our DSSINet consists of three parallel subnetworks with shared parameters, and each of them takes a different scaled version of the same image as input for feature extraction. Then, a unified Structured Feature Enhancement Module (SFEM) is proposed and integrated into our network for structured feature representation learning. Based on the *conditional random fields* (CRFs [16]), SFEM mutually refines the multiscale features from different subnetworks with a message passing mechanism [15]. Specifically, SFEM dynamically passes the complementary information from the features at other scales to enhance the scale-specific feature. Finally, we generate multiple side output density maps from the refined features and obtain a high-resolution density map in a top-down manner.

For the hierarchically structured loss optimization, we utilize a Dilated Multiscale Structural Similarity (DMS-SSIM) loss to enforce networks to learn the local correlation within regions of various sizes and produce locally consistent density maps. Specifically, our DMS-SSIM loss is designed for each pixel and it is computed by measuring the structural similarity between the multiscale regions centered at the given pixel on an estimated density map and the corresponding regions on ground-truth. Moreover, we implement the DMS-SSIM loss with a dilated convolution-



Figure 2. Illustration of the information of people’s scales on crowd density maps. The radiuses of people’s heads are relatively uniform in a local region. Moreover, the local correlation of people’s scales may vary in different regions.

al neural network, in which the dilated operation enlarges the diversity of the scales of local regions and can further improve the performance.

In summary, the contributions of our work are three-fold:

- We propose a CRFs-based Structured Feature Enhancement Module, which refines multiscale features mutually and boosts their robustness against scale variation by fully exploiting their complementarity.
- We utilize a Dilated Multiscale Structural Similarity loss to learn the local correlation within regions of various sizes. To our best knowledge, we are the first to incorporate the MS-SSIM [39] based loss function for crowd counting and verify its effectiveness in this task.
- Extensive experiments conducted on four challenging benchmarks demonstrate that our method achieves state-of-the-art performance.

2. Related Work

Crowd Counting: Numerous deep learning based methods [36, 30, 40, 19, 20, 23, 7] have been proposed for crowd counting. These methods have various network structures and the mainstream is a multiscale architecture, which extracts multiple features from different columns/branches of networks to handle the scale variation of people. For instance, Boominathan et al. [3] combined a deep network and a shallow network to learn scale-robust features. Zhang et al. [47] developed a multi-column CNN to generate density maps. HydraCNN [25] fed a pyramid of image patches into networks to estimate the count. CP-CNN [35] proposed a Contextual Pyramid CNN to incorporate the global and local contextual information for crowd counting. Cao et al. [4] built an encoder-decoder network with multiple scale aggregation modules. However, the issue of the huge variation of people’s scales is still far from being fully solved. *In this paper, we further strengthen the robustness of DSSINet against the scale variation of people from two aspects, including structured feature representation learning and hierarchically structured loss function optimization.*

Conditional Random Fields: In the field of computer vision, CRFs have been exploited to refine the features and outputs of convolutional neural networks (CNN) with a message passing mechanism [15]. For instance, Zhang et

¹In this paper, ground-truth crowd density maps are generated with geometry-adaptive Gaussian kernels [47]. Each person is marked as a Gaussian kernel with individual radius

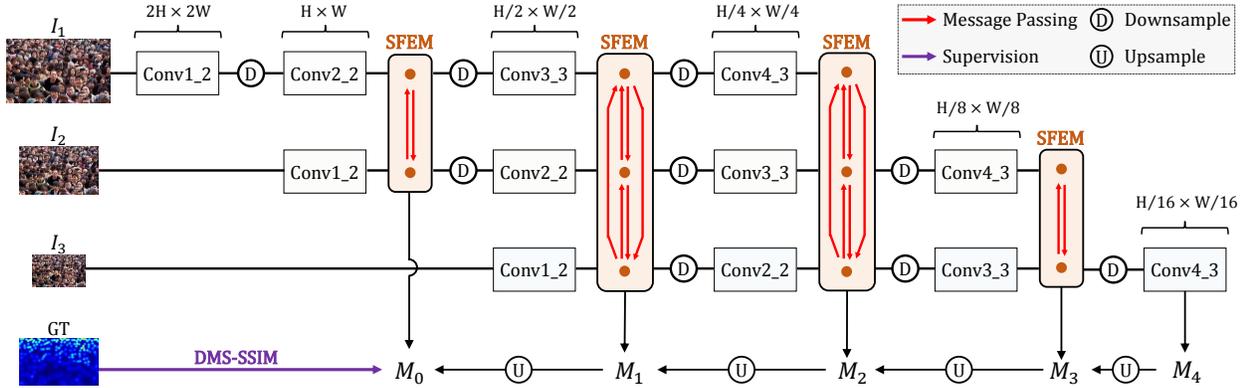


Figure 3. The overall framework of the proposed Deep Structured Scale Integration Network (DSSINet). DSSINet consists of three parallel subnetworks with shared parameters. These subnetworks take different scaled versions of the same image as input for feature extraction. First, DSSINet integrates four CRFs-based Structured Feature Enhancement Modules (SFEM) to refine the multiscale features from different subnetworks. Then, we progressively fuse multiple side output density maps to obtain a high-resolution one in a top-down manner. Finally, a Dilated Multiscale Structural Similarity (DMS-SSIM) loss is utilized to optimize our DSSINet.

al. [49] used CRFs to refine the semantic segmentation maps of CNN by modeling the relationship among pixels. Xu et al. [41] fused multiple features with Attention-Gated CRFs to produce richer representations for contour prediction. Wang et al. [37] introduced an inter-view message passing module based on CRFs to enhance the view-specific features for action recognition. *In crowd counting, we are the first to utilize CRFs to mutually refine multiple features at different scales and prove its effectiveness for this task.*

Multiscale Structural Similarity: MS-SSIM [39] is a widely used metric for image quality assessment. Its formula is based on the luminance, contrast and structure comparisons between the multiscale regions of two images. In [48], MS-SSIM loss has been successfully applied in image restoration tasks (e.g., image denoising and super-resolution), but its effectiveness has not been verified in high-level tasks (e.g. crowd counting). Recently, Cao et al. [4] combined Euclidean loss and SSIM loss [38] to optimize their network for crowd counting, but they can only capture the local correlation in regions with a fixed size. *In this paper, to learn the local correlation within regions of various sizes, we modify MS-SSIM loss with a dilated operation and show its effectiveness in this high-level task.*

3. Method

In this section, we propose a Deep Structured Scale Integration Network (DSSINet) for crowd counting. Specifically, it addresses the scale variation of people with structured feature representation learning and structured loss function optimization. For the former, a Structured Feature Enhancement Module based on *conditional random fields* (CRFs) is proposed to refine multiscale features mutually with a message passing mechanism. For the latter, a Dilated Multiscale Structural Similarity loss is utilized to enforce networks to learn the local correlation within regions of various sizes.

3.1. DSSINet Overview

In crowd counting, multiscale features are usually extracted to handle the scale variation of people. Inspired by [18, 13, 5], we build our DSSINet with three parallel subnetworks, which have the same architecture and share parameters. As shown in Fig. 3, these subnetworks are composed of the first ten convolutional layers (up to Conv4.3) of VGG16 [33] and each of them takes a different scaled version of the same image as input to extract features. Unlike previous works [47, 4] that simply fuse features by weighted averaging or concatenation, our DSSINet adequately refines the multiscale features from different subnetworks.

Given an image of size $H \times W$, we first build a three-levels image pyramid $\{I_1, I_2, I_3\}$, where $I_2 \in R^{H \times W}$ is the original image, and $I_1 \in R^{2H \times 2W}$ and $I_3 \in R^{\frac{H}{2} \times \frac{W}{2}}$ are the scaled ones. Each of these images is fed into one of the subnetworks. The feature of image I_k at the Conv $_{i,j}$ layer of VGG16 is denoted as $f_{i,j}^k$. We then group the features with the same resolution from different subnetworks and form four sets of multiscale features $\{f_{2,2}^1, f_{1,2}^2\}$, $\{f_{3,3}^1, f_{2,2}^2, f_{1,2}^3\}$, $\{f_{4,3}^1, f_{3,3}^2, f_{2,2}^3\}$, $\{f_{4,3}^2, f_{3,3}^3\}$. In each set, different features complement each other, because they are inferred from different receptive fields and are derived from different convolutional layers of various image sizes. For instance, $f_{1,2}^2$ mainly contains the appearance details and $f_{2,2}^1$ encodes some high-level semantic information. To improve the robustness of scale variation, we refine the features in the aforementioned four sets with the Structured Feature Enhancement Module described in Section 3.2. With richer information, the enhanced feature $\hat{f}_{i,j}^k$ of $f_{i,j}^k$ becomes more robust to the scale variation. Then, $\hat{f}_{i,j}^k$ is fed into the following layer of k^{th} subnetwork for deeper feature representation learning.

After structured feature learning, we generate a high-

resolution density map in a top-down manner. First, we apply a 1×1 convolutional layer on top of the last feature $f_{4,3}^3$ for reducing its channel number to 128, and then feed the compressed feature into a 3×3 convolutional layer to generate a density map M_4 . However, with a low resolution of $\frac{H}{16} \times \frac{W}{16}$, M_4 lacks spatial information of people. To address this issue, we generate other four side output density maps $\tilde{M}_0, \tilde{M}_1, \tilde{M}_2, \tilde{M}_3$ at shallower layers, where \tilde{M}_i has a resolution of $\frac{H}{2^i} \times \frac{W}{2^i}$. Specifically, \tilde{M}_3 is computed by feeding the concatenation of $f_{4,3}^2$ and $f_{3,3}^3$ into two stacked convolutional layers. The first 1×1 convolutional layer is also utilized to reduce the channel number of the concatenated feature to 128, while the second 3×3 convolutional layer is used to regress \tilde{M}_3 . $\tilde{M}_2, \tilde{M}_1, \tilde{M}_0$ are obtained in the same manner. Finally, we progressively pass the density maps at deeper layers to refine the density maps at shallower layers and the whole process can be expressed as:

$$M_i = w_i * \tilde{M}_i + w_{i+1} * Up(M_{i+1}), i = 3, 2, 1, 0 \quad (1)$$

where w_i and w_{i+1} are the parameters of two 3×3 convolutional layers and $Up()$ denotes a bilinear interpolation operation with a upsampling rate of 2. M_i is the refined density map of \tilde{M}_i . The final crowd density map $M_0 \in R^{H \times W}$ has fine details of the spatial distribution of people.

Finally, we train our DSSINet with the Dilated Multiscale Structural Similarity (DMS-SSIM) loss described in Section 3.3. We implement DMS-SSIM loss with a lightweight dilated convolutional neural network with fixed Gaussian-kernel and the gradient can be back-propagated to optimize our DSSINet.

3.2. Structured Feature Enhancement Module

In this subsection, we propose a unified Structured Feature Enhancement Module (SFEM) to improve the robustness of our feature for scale variation. Inspired by the dense prediction works [6, 41], our SFEM mutually refines the features at different scales by fully exploring their complementarity with a *conditional random fields* (CRFs) model. In this module, each scale-specific feature passes its own information to features at other scales. Meanwhile, each feature is refined by dynamically fusing the complementary information received from other features.

Let us denote multiple features extracted from different subnetworks as $F = \{f_1, f_2, \dots, f_n\}$. F can be any of the multiscale features sets defined in Section 3.1. Our objective is to estimate a group of refined features $\hat{F} = \{\hat{f}_1, \hat{f}_2, \dots, \hat{f}_n\}$, where \hat{f}_i is the corresponding refined feature of f_i . We formulate this problem with a CRFs model. Specifically, the conditional distribution of the original feature F and the refined feature \hat{F} is defined as:

$$P(\hat{F}|F, \Theta) = \frac{1}{Z(F)} \exp\{E(\hat{F}, F, \Theta)\}, \quad (2)$$

where $Z(F) = \int_{\hat{F}} \exp\{E(\hat{F}, F, \Theta)\} d\hat{F}$ is the partition function for normalization and Θ is the set of parameters.

The energy function $E(\hat{F}, F, \Theta)$ in CRFs is defined as:

$$E(\hat{F}, F, \Theta) = \sum_i \phi(\hat{f}_i, f_i) + \sum_{i,j} \psi(\hat{f}_i, \hat{f}_j). \quad (3)$$

In particular, the unary potential $\phi(\hat{f}_i, f_i)$, indicating the similarity between the original feature and the refined feature, is defined as:

$$\phi(\hat{f}_i, f_i) = -\frac{1}{2} \|\hat{f}_i - f_i\|^2. \quad (4)$$

We model the correlation between two refined features with a bilinear potential function, thus the pairwise potential is defined as:

$$\psi(\hat{f}_i, \hat{f}_j) = (\hat{f}_i)^T w_j^i \hat{f}_j, \quad (5)$$

where w_j^i is a learned parameter used to compute the relationship between \hat{f}_i and \hat{f}_j .

This is a typical formulation of CRF and we solve it with mean-field inference [29]. The feature \hat{f}_i is computed by:

$$\hat{f}_i = f_i + \sum_{j \neq i} w_j^i \hat{f}_j, \quad (6)$$

where the unary term is feature f_i itself and the second term denotes the information received from other features at different scales. The parameter w_j^i determines the information content passed from f_j to f_i . As \hat{f}_i and \hat{f}_j are interdependent in Eq.(6), we obtain each refined feature iteratively with the following formulation:

$$h_i^0 = f_i, \quad h_i^t = f_i + \sum_{j \neq i} w_j^i h_j^{t-1}, t = 1 \text{ to } n, \quad \hat{f}_i = h_i^n, \quad (7)$$

where n is the total iteration number and h_i^t is the intermediate feature at t^{th} iteration. The Eq.(7) can be easily implemented in our SFEM. Specifically, we apply a 1×1 convolutional layer to pass the complementary information from f_j to f_i . w_j^i is the learned parameter of the convolutional layer and it is shared for all iterations.

As shown in Fig. 3, we apply the proposed SFEM to mutually refine the features in $\{f_{2,2}^1, f_{1,2}^2\}$, $\{f_{3,3}^1, f_{2,2}^2, f_{1,2}^3\}$, $\{f_{4,3}^1, f_{3,3}^2, f_{2,2}^3\}$, $\{f_{4,3}^2, f_{3,3}^3\}$. After receiving the information from other features at different scales, the refined feature becomes more robust to the scale variation of people. The experiments in Section 4 show that our SFEM greatly improves the performance of crowd counting.

3.3. Dilated Multiscale Structural Similarity Loss

In this subsection, we employ a Dilated Multiscale Structural Similarity (DMS-SSIM) loss to train our network. We ameliorate the original MS-SSIM [39] with dilation operations to enlarge the diversity of the sizes of local regions and force our network to capture the local correlation within regions of different sizes. Specifically, for each pixel, our DMS-SSIM loss is computed by measuring the structural similarity between the multiscale regions centered at the given pixel on an estimated density map and the corresponding regions on the GT density map.

As shown in Fig. 4, we implement the DMS-SSIM loss

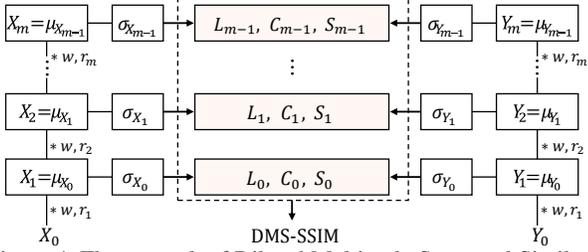


Figure 4. The network of Dilated Multiscale Structural Similarity loss. The normalized Gaussian kernel w is fixed and shared for all layers. r_i is the dilation rate of the i^{th} layer. X_0 and Y_0 are the estimated density map and the corresponding GT map respectively.

with a dilated convolutional neural network named as DMS-SSIM network. The DMS-SSIM network consists of m dilated convolutional layers and the parameters of these layers are fixed as a normalized Gaussian kernel with a size of 5×5 and a standard deviation of 1.0. The Gaussian kernel is denoted as $w = \{w(o) | o \in O, O = \{(-2, -2), \dots, (2, 2)\}\}$, where o is an offset from the center.

For convenience, the estimated density map M_0 in DSSINet is re-marked as X_0 in this subsection and its corresponding ground-truth is denoted as Y_0 . We feed X_0 and Y_0 into the DMS-SSIM network respectively and their outputs at i^{th} layer are represented as $X_i \in R^{H \times W}$ and $Y_i \in R^{H \times W}$. Specifically, for a given location p , $X_{i+1}(p)$ is calculated by:

$$X_{i+1}(p) = \sum_{o \in O} w(o) \cdot X_i(p + r_{i+1} \cdot o), \quad (8)$$

where r_{i+1} is the dilation rate of the $(i+1)^{th}$ layer and it is used to control the size of receptive field. Since $\sum_{o \in O} w(o) = 1$, $X_{i+1}(p)$ is the weighted mean of a local region centered at the location p on X_i and we could get $X_{i+1} = \mu_{X_i}$. In this case, X_1 is the local mean of X_0 and X_2 is the local mean of X_1 . By analogy, $X_{i+1}(p)$ can be considered as the mean of a relatively large region on X_0 . Based on the filtered map X_{i+1} , we calculate the local variance $\sigma_{X_i}^2(p)$ as:

$$\sigma_{X_i}^2(p) = \sum_{o \in O} w(o) \cdot [X_i(p + r_{i+1} \cdot o) - \mu_{X_i}(p)]^2. \quad (9)$$

The local mean μ_{Y_i} and variance $\sigma_{Y_i}^2$ of filtered map Y_i are also calculated with the same formulations as Eq.(8) and Eq.(9). Moreover, the local covariance $\sigma_{X_i Y_i}^2(p)$ between X_i and Y_i can be computed by:

$$\sigma_{X_i Y_i}^2(p) = \sum_{o \in O} \{w(o) \cdot [X_i(p + r_{i+1} \cdot o) - \mu_{X_i}(p)] \cdot [Y_i(p + r_{i+1} \cdot o) - \mu_{Y_i}(p)]\} \quad (10)$$

Further, the luminance comparison L_i , contrast comparison C_i and structure comparison S_i between X_i and Y_i are formulated as:

$$L_i = \frac{2\mu_{X_i}\mu_{Y_i} + c_1}{\mu_{X_i}^2 + \mu_{Y_i}^2 + c_1}, C_i = \frac{2\sigma_{X_i}\sigma_{Y_i} + c_2}{\sigma_{X_i}^2 + \sigma_{Y_i}^2 + c_2}, S_i = \frac{\sigma_{X_i Y_i} + c_3}{\sigma_{X_i}\sigma_{Y_i} + c_3}, \quad (11)$$

where c_1 , c_2 and c_3 are small constants to avoid division by

Layer	1 st	2 nd	3 rd	4 th	5 th
W/O Dilation	5 (1)	9 (1)	13 (1)	17 (1)	21 (1)
W/- Dilation	5 (1)	13 (2)	25 (3)	49 (6)	85 (9)

Table 1. Comparison of the receptive field in DMS-SSIM network with and without dilation. The receptive field f_i and dilation rate r_i of i^{th} layer are denoted as “ $f_i(r_i)$ ” in the table. Without dilation ($r_i = 1$ for all layers), the receptive fields of all layers are clustered into a small region, which makes DMS-SSIM network fail to capture the local correlation in regions with large heads.

zero. The SSIM between X_i and Y_i is calculated as:

$$SSIM(X_i, Y_i) = L_i \cdot C_i \cdot S_i. \quad (12)$$

Finally, the DMS-SSIM between the estimated density map X_0 and ground-truth Y_0 , as well as the DMS-SSIM loss are defined as:

$$DMS-SSIM(X_0, Y_0) = \prod_{i=0}^{m-1} \{SSIM(X_i, Y_i)\}^{\alpha_i}, \quad (13)$$

$$Loss(X_0, Y_0) = 1 - DMS-SSIM(X_0, Y_0)$$

where α_i is the importance weight of $SSIM(X_i, Y_i)$ and we refer to [39] to set the value of α_i .

In this study, the number of layers in DMS-SSIM network m is set to 5. The dilation rates of these layers are 1, 2, 3, 6 and 9 respectively and we show the receptive field of each layer in Table 1. The receptive field of the 2^{nd} layer is 13, thus $SSIM(X_1, Y_1)$ indicates the local similarity measured in regions with size of 13×13 . Similarly, $SSIM(X_4, Y_4)$ is the local similarity measured in 85×85 regions. When removing the dilation, the receptive fields of all layers are clustered into a small region, which makes DMS-SSIM network fail to capture the local correlation in regions with large heads, such as the green box in Fig. 2. The experiments in Section 4.4 show that the dilation operation is crucial to obtain the accurate count of crowd.

4. Experiments

4.1. Implementation Details

Ground-Truth Density Maps Generation: In this work, we generate ground-truth density maps with the geometry-adaptive Gaussian kernels [47]. For each head annotation p_i , we mark it as a Gaussian kernel $\mathcal{N}(p_i, \sigma^2)$ on the density map, where the spread parameter σ is equal to 30% of the mean distance to its three nearest neighbors. Moreover, the kernel is truncated within 3σ and normalized to an integral of 1. Thus, the integral of the whole density map is equal to the crowd count in the image.

Networks Optimization: Our framework is implemented with PyTorch [26]. We use the first ten convolutional layers of the pre-trained VGG-16 to initialize the corresponding convolution layers in our framework. The rest of convolutional layers are initialized by a Gaussian distribution with zero mean and standard deviation of $1e-6$. At each training

Method	Part A		Part B	
	MAE	MSE	MAE	MSE
MCNN [47]	110.2	173.2	26.4	41.3
SwitchCNN [30]	90.4	135	21.6	33.4
CP-CNN [35]	73.6	106.4	20.1	30.1
DNCL [32]	73.5	112.3	18.7	26.0
ACSCP [31]	75.7	102.7	17.2	27.4
IG-CNN [1]	72.5	118.2	13.6	21.1
IC-CNN [28]	68.5	116.2	10.7	16.0
CSRNet [17]	68.2	115.0	10.6	16.0
SANet [4]	67.0	104.5	8.4	13.6
Ours	60.63	96.04	6.85	10.34

Table 2. Performance comparison on Shanghaitech dataset.

iteration, 16 image patches with a size of 224×224 are randomly cropped from images and fed into DSSINet. We optimize our network with Adam [14] and a learning rate of $1e-5$ by minimizing the DMS-SSIM loss.

4.2. Evaluation Metric

For crowd counting, *Mean Absolute Error* (MAE) and *Mean Squared Error* (MSE) are two metrics widely adopted to evaluate the performance. They are defined as follows:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N \|\hat{P}_i - P_i\|, \text{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\hat{P}_i - P_i\|^2}, \quad (14)$$

where N is the total number of the testing images, \hat{P}_i and P_i are the estimated count and the ground truth count of the i^{th} image respectively. Specifically, \hat{P}_i is calculated by the integration of the estimated density map.

4.3. Comparison with the State of the Art

Comparison on Shanghaitech [47]: As the most representative benchmark of crowd counting, Shanghaitech dataset contains 1,198 images with a total of 330 thousand annotated people. This dataset can be further divided into two parts: Part A with 482 images randomly collected from the Internet and Part B with 716 images taken from a busy shopping street in Shanghai, China.

We compare the proposed method with ten state-of-the-art methods on this dataset. As shown in Table 2, our method achieves superior performance on both parts of the Shanghaitech dataset. Specifically, on Part A, our method achieves a relative improvement of 9.5% in MAE and 8.1% in MSE over the existing best algorithm SANet [4]. Although previous methods have worked well on Part B, our method still achieves considerable performance gain by decreasing the MSE from 13.6 to 10.34. The visualization results in Fig. 5 show that our method can generate high-quality crowd density maps with accurate counts, even though the scales of people vary greatly in images.

Comparison on UCF-QNRF [12]: The recently released UCF-QNRF dataset is a challenging benchmark for dense crowd counting. It consists of 1,535 unconstrained

Method	MAE	MSE
Idrees et al. [11]	315	508
MCNN [47]	277	426
Encoder-Decoder [2]	270	478
CMTL [34]	252	514
SwitchCNN [30]	228	445
Resnet-101 [9]	190	277
Densenet-201 [10]	163	226
CL [12]	132	191
Ours	99.1	159.2

Table 3. Performance of different methods on UCF-QNRF dataset.

Method	MAE	MSE
MCNN [47]	377.6	509.1
SwitchCNN [30]	318.1	439.2
CP-CNN [35]	295.8	320.9
IG-CNN [1]	291.4	349.4
ConvLSTM [40]	284.5	297.1
CSRNet [17]	266.1	397.5
IC-CNN [28]	260.9	365.5
SANet [4]	258.4	334.9
Ours	216.9	302.4

Table 4. Performance comparison on UCF_CC_50. The results of top two performance are highlighted in red and blue respectively.

crowded images (1,201 for training and 334 for testing) with huge scale, density and viewpoint variations. 1.25 million persons are annotated and they are unevenly dispersed to images, varying from 49 to 12,865 per image.

On UCF-QNRF dataset, we compare our DSSINet with eight methods, including Idrees et al. [11], MCNN [47], Encoder-Decoder [2], CMTL [34], SwitchCNN [30], Resnet-101 [9], Densenet-201 [10] and CL [12]. The performances of all methods are summarized in Table 3 and we can observe that our DSSINet exhibits the lowest MAE 99.1 and MSE 159.2 on this dataset, outperforming other methods with a large margin. Specifically, our method achieves a significant improvement of 24.9% in MAE over the existing best-performing algorithm CL.

Comparison on UCF_CC_50 [11]: This is an extremely challenging dataset. It contains 50 crowded images of various perspective distortions. Moreover, the number of people varies greatly, ranging from 94 to 4,543. Following the standard protocol in [11], we divide the dataset into five parts randomly and perform five-fold cross-validation. We compare our DSSINet with thirteen state-of-the-art methods on this dataset. As shown in Table 4, our DSSINet obtains a MAE of 216.9 and outperforms all other methods. Specifically, our method achieves a relative improvement of 19.1% in MAE over the existing best algorithm SANet [4].

Comparison on WorldExpo'10 [43]: As a large-scale crowd counting benchmark with the largest amount of images, WorldExpo'10 contains 1,132 video sequences cap-

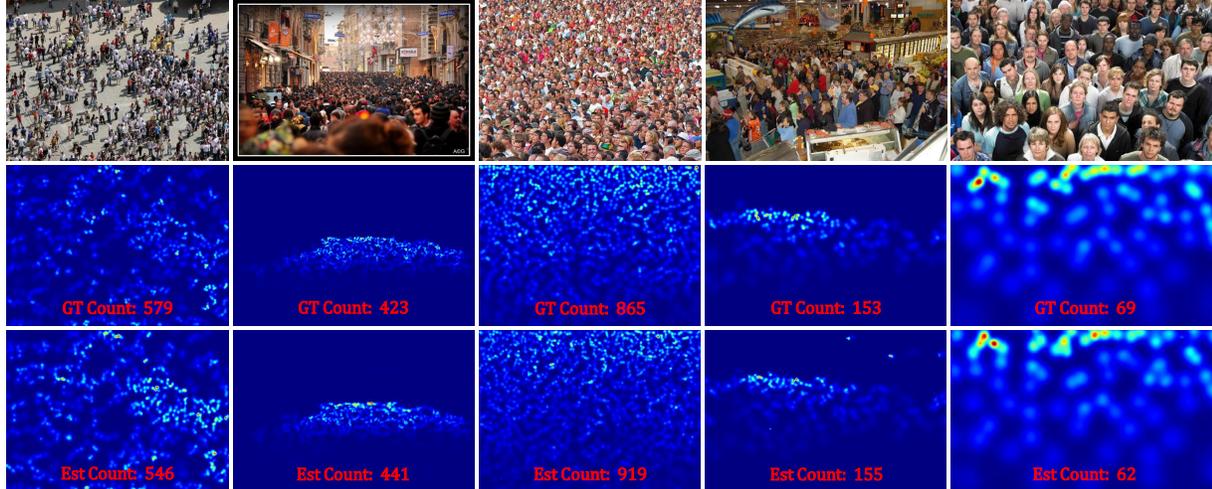


Figure 5. Visualization of the crowd density maps generated by our method on ShanghaiTech Part A. The first row shows the testing images with people of various scales. The second row shows the ground-truth density maps and the standard counts, while the third row presents our generated density maps and estimated counts. Our method can generate high-quality crowd density maps with accurate counts, even though the scales of people vary greatly in images.

Method	S1	S2	S3	S4	S5	Ave
Zhang et al [43]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN [47]	3.4	20.6	12.9	13.0	8.1	11.6
ConvLSTM [40]	7.1	15.2	15.2	13.9	3.5	10.9
SwitchCNN [30]	4.4	15.7	10.0	11.0	5.9	9.4
DNCL [32]	1.9	12.1	20.7	8.3	2.6	9.1
CP-CNN [35]	2.9	14.7	10.5	10.4	5.8	8.86
CSRNet [17]	2.9	11.5	8.6	16.6	3.4	8.6
SANet [4]	2.6	13.2	9.0	13.3	3.0	8.2
DRSAN [20]	2.6	11.8	10.3	10.4	3.7	7.76
ACSCP [31]	2.8	14.05	9.6	8.1	2.9	7.5
Ours	1.57	9.51	9.46	10.35	2.49	6.67

Table 5. MAE of different methods on the WorldExpo’10 dataset.

tured by 108 surveillance cameras during the Shanghai WorldExpo 2010. Following the standard protocol in [43], we take 3,380 annotated frames from 103 scenes as training set and 600 frames from remaining five scenes as testing set. When testing, we only measure the crowd count under the given Region of Interest (RoI).

The mean absolute errors of our method and thirteen state-of-the-art methods are summarized in Table 5. Our method exhibits the lowest MAE in three scenes and achieves the best performance with respect to the average MAE of five scenes. Moreover, compared with those methods that rely on temporal information [40] or perspective map [43], our method is more flexible to generate density maps and estimate the crowd counts.

4.4. Ablation Study

Effectiveness of Structured Feature Enhancement Module: To validate the effectiveness of SFEM, we implement the following variants of our DSSINet:

- **W/O FeatRefine:** This model feeds the image pyramid $\{I_1, I_2, I_3\}$ into the three subnetworks, but it doesn’t conduct feature refinement. It takes the original features to generate side output density maps. For example, \tilde{M}_1 is directly generated from $f_{3,3}^1, f_{2,2}^2$ and $f_{1,2}^3$.
- **ConcatConv FeatRefine:** This model also takes $\{I_1, I_2, I_3\}$ as input and it attempts to refine multiscale features with concatenation and convolution. For instance, it feeds the concatenation of $f_{3,3}^1, f_{2,2}^2, f_{1,2}^3$ into a 1×1 convolutional layer to compute the feature $\hat{f}_{3,3}^1$. $\hat{f}_{2,2}^2$ and $\hat{f}_{1,2}^3$ are obtained in the same manner.
- **CRF-n FeatRefine:** This model uses the proposed CRFs-based SEFM to refine multiscale features from $\{I_1, I_2, I_3\}$. We explore the influence of the iteration number n in CRF, e.g., $n=1,2,3$.

We train and evaluate all aforementioned variants on ShanghaiTech Part A. As shown in Table 6, the variant “W/O FeatRefine” obtains the worst performance for the lack of features refinement. Although “ConcatConv FeatRefine” can reduce the count error to some extent by simply refining multiple features, its performance is still barely satisfactory. In contrast, our SFEM fully exploits the complementarity among multiscale features and mutually refines them with CRFs, and thus significantly boosts the performance. Specifically, our “CRF-2 FeatRefine” achieves the best performance. However, too longer iteration number of CRFs in SFEM would degrade the performance (See “CRF-3 FeatRefine”), since those multiscale features may be excessively mixed and loss their own semantic meanings. Thus, the iteration number n in CRFs is 2 in our final model.

Influence of the Scale Number of Image Pyramid: To validate the effectiveness of multiscale input, we train mul-

Method	MAE	MSE
W/O FeatRefine	68.85	119.09
ConcatConv FeatRefine	67.11	110.87
CRF-1 FeatRefine	64.37	108.61
CRF-2 FeatRefine	60.63	96.04
CRF-3 FeatRefine	63.80	103.05

Table 6. Performance of different variants of DSSINet on Part A of Shanghaitech dataset.

Scales	1	1+0.5	2+1+0.5	2+1+0.5+0.25
MAE	70.94	64.67	60.63	61.13

Table 7. Ablation study of the scale number of image pyramid on Part A of Shanghaitech dataset.

multiple variants of DSSINet with different scale number of image pyramid and summarize their performance in Table 7. Notice that the single-scale variant directly feeds the given original image into one subnetwork to extract features and generates the final density map from four side output density maps at Conv1.2, Conv2.2, Conv3.3 and Conv4.3. The term “2+1+0.5” denotes an image pyramid with scales 2, 1 and 0.5. Other terms can be understood by analogy. We can observe that the performance gradually increases as the scale number increases and it is optimal with three scales. Since the computation was too large when the scale ratio was set to 4 or larger, we did not include more.

Effectiveness of Dilated Multiscale Structural Similarity Loss: In this subsection, we evaluate the effectiveness of the DMS-SSIM loss for crowd counting. For the purpose of comparison, we train multiple variants of DSSINet with different loss functions, including Euclidean loss, SSIM loss and various configurations of DMS-SSIM loss. Note that a DMS-SSIM loss with m scales is denoted as “DMS-SSIM- m ” and its simplified version without dilation is denoted as “MS-SSIM- m ”. The performances of all loss functions are summarized in Table 8. We can observe that the performance of DMS-SSIM loss gradually improves, as the scale number m increases. When adopting “DMS-SSIM-5”, our DSSINet achieves the best MAE 60.63 and MSE 96.04, outperforming the models trained by Euclidean loss or SSIM loss. We also implement a “DMS-SSIM-6” loss, in which the sixth layer has a dilation rate of 9 and it attempts to capture the local correlation in 121×121 regions. However, the people’s scales may not uniform in such large regions, thus the performance of “DMS-SSIM-6” has slightly dropped, compared with “DMS-SSIM-5”. Moreover, the performance of MS-SSIM loss is worse than that of DMS-SSIM loss, since the receptive fields in MS-SSIM loss are intensively clustered into a small region, which makes our DSSINet fail to learn the local correlation of the people with various scales. These experiments well demonstrate the effectiveness of DMS-SSIM loss.

Complexity Analysis: We also discuss the complexity

Loss Function	MAE	MSE
Euclidean	67.68	108.45
SSIM	74.60	133.64
MS-SSIM-2	73.21	125.05
MS-SSIM-3	67.46	114.79
MS-SSIM-4	64.80	109.26
MS-SSIM-5	63.51	103.81
DMS-SSIM-2	73.33	121.87
DMS-SSIM-3	67.12	112.86
DMS-SSIM-4	62.90	105.14
DMS-SSIM-5	60.63	96.04
DMS-SSIM-6	62.60	103.27

Table 8. Performance evaluation of different loss functions on Part A of Shanghaitech dataset. “DMS-SSIM- m ” denotes a DMS-SSIM loss with m scales and “MS-SSIM- m ” is the corresponding simplified version without dilation.

Model	CP-CNN	SwitchCNN	CSRNet	Ours
Parameter	68.4	15.11	16.26	8.85

Table 9. Comparison of the number of parameters (in millions).

of our method. As the subnetworks in our framework have shared parameters and the kernel size of the convolutional layers in SFEM is 1×1 , the proposed DSSINet only has 8.858 million parameters, 86.19% (7.635 million) of which come from its backbone network (the first ten convolutional layers of VGG-16). As listed in Table 9, the number of parameters of our DSSINet is only half of that of the existing state-of-the-arts (e.g. CSRNet). Compared with these methods, our DSSINet achieves better performance with much fewer parameters. During the testing phase, DSSINet takes 450 ms to process a 720×576 frame from SD surveillance videos on an NVIDIA 1080 GPU. This runtime speed is already qualified for the needs of many practical applications, since people do not move so fast and not every frame needs to be analyzed.

5. Conclusion

In this paper, we develop a Deep Structured Scale Integration Network for crowd counting, which handles the huge variation of people’s scales from two aspects, including structured feature representation learning and hierarchically structured loss function optimization. First, a Structured Feature Enhancement Module based on *conditional random fields* (CRFs) is proposed to mutually refine multiple features and boost their robustness. Second, we utilize a Dilated Multiscale Structural Similarity Loss to force our network to learn the local correlation within regions of various sizes, thereby producing locally consistent estimation results. Extensive experiments on four benchmarks show that our method achieves superior performance in comparison to the state-of-the-art methods.

References

- [1] Deepak Babu Sam, Neeraj N Sajjan, R Venkatesh Babu, and Mukundhan Srinivasan. Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn. In *CVPR*, pages 3618–3626, 2018.
- [2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.
- [3] Lokesh Boominathan, Srinivas SS Kruthiventi, and R Venkatesh Babu. Crowdnet: A deep convolutional network for dense crowd counting. In *ACM MM*, pages 640–644. ACM, 2016.
- [4] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *ECCV*, pages 734–750, 2018.
- [5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 40(4):834–848, 2018.
- [6] Xiao Chu, Wanli Ouyang, Xiaogang Wang, et al. Crf-cnn: Modeling structured information in human pose estimation. In *NIPS*, pages 316–324, 2016.
- [7] Junyu Gao, Qi Wang, and Xuelong Li. Pcc net: Perspective crowd counting via spatial convolutional network. *TCSVT*, 2019.
- [8] Weina Ge and Robert T Collins. Marked point processes for crowd counting. In *CVPR*, pages 2913–2920. IEEE, 2009.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [10] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.
- [11] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. Multi-source multi-scale counting in extremely dense crowd images. In *CVPR*, pages 2547–2554, 2013.
- [12] Haroon Idrees, Muhammad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *ECCV*, 2018.
- [13] Di Kang and Antoni Chan. Crowd counting by adaptively fusing predictions from an image pyramid. In *BMVC*, 2018.
- [14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.
- [15] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.
- [16] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.
- [17] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *CVPR*, pages 1091–1100, 2018.
- [18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017.
- [19] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G Hauptmann. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In *CVPR*, pages 5197–5206, 2018.
- [20] Lingbo Liu, Hongjun Wang, Guanbin Li, Wanli Ouyang, and Liang Lin. Crowd counting using deep recurrent spatial-aware network. In *IJCAI*, 2018.
- [21] Lingbo Liu, Ruimao Zhang, Jiefeng Peng, Guanbin Li, Bowen Du, and Liang Lin. Attentive crowd flow machines. In *ACM MM*, pages 1553–1561. ACM, 2018.
- [22] Ning Liu, Yongchao Long, Changqing Zou, Qun Niu, Li Pan, and Hefeng Wu. Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding. *arXiv preprint arXiv:1811.11968*, 2018.
- [23] Weizhe Liu, Krzysztof Maciej Lis, Mathieu Salzmann, and Pascal Fua. Geometric and physical constraints for drone-based head plane crowd density estimation. In *IROS*, 2019.
- [24] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. *arXiv preprint arXiv:1811.10452*, 2018.
- [25] Daniel Onoro-Rubio and Roberto J López-Sastre. Towards perspective-free object counting with deep learning. In *EC-CV*, pages 615–629. Springer, 2016.
- [26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- [27] Zhilin Qiu, Lingbo Liu, Guanbin Li, Qing Wang, Nong Xiao, and Liang Lin. Crowd counting via multi-view scale aggregation networks. In *ICME*, 2019.
- [28] Viresh Ranjan, Hieu Le, and Minh Hoai. Iterative crowd counting. In *ECCV*, 2018.
- [29] Kosta Ristovski, Vladan Radosavljevic, Slobodan Vucetic, and Zoran Obradovic. Continuous conditional random fields for efficient regression in large fully connected graphs. In *AAAI*, 2013.
- [30] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. Switching convolutional neural network for crowd counting. In *CVPR*, volume 1, page 6, 2017.
- [31] Zan Shen, Yi Xu, Bingbing Ni, Minsi Wang, Jianguo Hu, and Xiaokang Yang. Crowd counting via adversarial cross-scale consistency pursuit. In *CVPR*, pages 5245–5254, 2018.
- [32] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, Ming-Ming Cheng, and Guoyan Zheng. Crowd counting with deep negative correlation learning. In *CVPR*, pages 5382–5390, 2018.
- [33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [34] Vishwanath A Sindagi and Vishal M Patel. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *AVSS*, pages 1–6. IEEE, 2017.

- [35] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *ICCV*, pages 1879–1888. IEEE, 2017.
- [36] Elad Walach and Lior Wolf. Learning to count with cnn boosting. In *ECCV*, pages 660–676. Springer, 2016.
- [37] Dongang Wang, Wanli Ouyang, Wen Li, and Dong Xu. Dividing and aggregating network for multi-view action recognition. In *ECCV*, pages 451–467, 2018.
- [38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.
- [39] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multi-scale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1398–1402. Ieee, 2003.
- [40] Feng Xiong, Xingjian Shi, and Dit-Yan Yeung. Spatiotemporal modeling for crowd counting in videos. In *ICCV*. IEEE, 2017.
- [41] Dan Xu, Wanli Ouyang, Xavier Alameda-Pineda, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. In *NIPS*, pages 3961–3970, 2017.
- [42] Lingke Zeng, Xiangmin Xu, Bolun Cai, Suo Qiu, and Tong Zhang. Multi-scale convolutional neural networks for crowd counting. In *ICIP*, pages 465–469. IEEE, 2017.
- [43] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*, pages 833–841, 2015.
- [44] Lu Zhang, Ju Dai, Huchuan Lu, You He, and Gang Wang. A bi-directional message passing model for salient object detection. In *CVPR*, pages 1741–1750, 2018.
- [45] Lu Zhang, Miaojing Shi, and Qiaobo Chen. Crowd counting via scale-adaptive convolutional neural network. In *WACV*. IEEE, 2018.
- [46] Shanghang Zhang, Guanhang Wu, Joao P Costeira, and Jose MF Moura. Understanding traffic density from large-scale web camera data. In *CVPR, ICME2017*.
- [47] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *CVPR*, pages 589–597, 2016.
- [48] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *TCI*, 2017.
- [49] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *ICCV*, pages 1529–1537, 2015.