Scale-aware Crowd Counting via Depth-embedded Convolutional Neural Networks

Muming Zhao, Chongyang Zhang, *Member, IEEE,* Jian Zhang, *Senior Member, IEEE,* Fatih Porikli, *Fellow, IEEE,* Bingbing Ni, and Wenjun Zhang, *Fellow, IEEE*

***Abstract*—Scale variation of pedestrians in a crowd image presents a significant challenge for vision-based people counting systems. Such variations are mainly caused by perspective-related distortions due to the camera pose relative to the ground plane. Following the density-based counting paradigm, we postulate that generating density values adaptive to object scales plays a critical role in the accuracy of the final counting results. Motivated by this, we distill the underlying information from depth cues to obtain scale-aware representations that can respond to object scales considering the fact that the scale is inversely proportional to the object depth. Specifically, we propose a depth embedding module as add-ons into existing networks. This module exploits essential depth cues to spatially re-calibrate the magnitude of the original features. In this way, the objects, although in the same class, will attain distinct representations according to their scales, which directly benefits the estimation of scale-aware density values. We conduct a comprehensive analysis of the effects of the depth embedding module and validate that exploiting depth cues to perceive object scale variations in convolutional neural networks improves crowd counting performances. Our experiments demonstrate the effectiveness of the proposed approach on four popular benchmark datasets.**

***Index Terms*—Crowd counting, depth embedding, perspective distortion, scale variation**

I. Introduction

ESTIMATING the number of people in crowded scenes is an essential task in a wide spectrum of video surveillance applications such as physical security, public space management, and retail space design [1]. It also provides an indispensable cue for higher-level scene understanding tasks including crowd behavior analysis and surveillance event recognition [2], [3]. These practical ramifications have brought increasing attention to visual crowd counting research in the recent past.

Manuscript received April 30, 2019; revised August 17, 2019. This work was partially funded by the National Science Fund of China under Grant No.61571297 and No.61420106008, the National Key Research and Development Program No.2017YFB1002401, 111 Program No.B07022 and STCSM No.18DZ2270700 and No.18DZ1112300. (*Corresponding author: Chongyang Zhang, sunny_zhang@sjtu.edu.cn)*

M. Zhao is with the Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China, and also is a joint-degree PhD student with the Faculty of Engineering and IT, University of Technology Sydney, Sydney, NSW 2007, Australia.

C. Zhang, B. Ni and W. Zhang are with the Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China.

J. Zhang is with the Faculty of Engineering and IT, University of Technology Sydney, Sydney, NSW 2007, Australia.

Fatih Porikli is with the Research School of Engineering, Australian National University (ANU), Canberra, Australia.

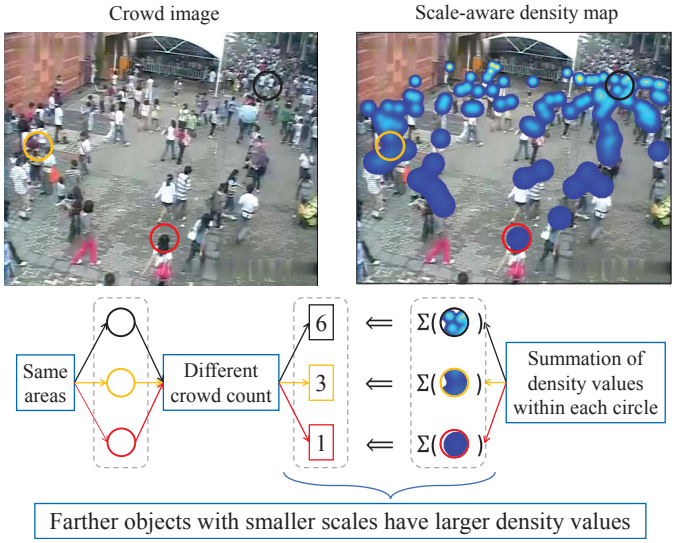


Fig. 1. Motivation (best viewed in color): Due to scale changes of pedestrians, the three regions (black, orange and red circles) that occupy the same number of pixels have different crowd counts; six in the far field (black), three in the midway (orange), and one in the near field (red) respectively. Since these three regions have the same area, the density values within the farthest circle should be larger than the ones in the nearer circles. In other words, objects with smaller scales should have larger density values and vice versa. This can be interpreted as *scale-aware* density values.

Due to magnifications and perspective-related distortions [4], images depicting crowed scenes often contain people with extreme scale variations, posing great challenges for general counting systems that operate on uncalibrated camera systems. In Fig. 1, we show a sample crowd image where objects closer to the camera appear significantly larger compared to the objects at farther distances. Roughly speaking, the scales of objects are inversely proportional to their distances to the camera imaging plane [5].

The popular density-based approaches [1], [6] generally determine crowd count by summation of the density values over specific regions in an estimated 2D density map. Following this paradigm, we postulate that a counting model should compensate for the object scale variations and work on *scale-aware* density values to achieve accurate estimates. As illustrated in Fig. 1, three different regions in the crowd image (marked with black, orange, and red circles) contain the same number of pixels. However, due to perspective-related distortions, each region contains a different number of people, i.e., there are six, three, and one pedestrian in the farthest

(black), medium (orange), and the nearest (red) circle, respectively. Since three regions have the same area, the density values at these three positions should vary accordingly to generate the correct estimates when we sum the density values over each region. More specifically, the density values in the farthest circle should be larger than those in the nearer regions. This suggests that a counting system should infer *scale-aware* density values and compensate for the scale variations caused by magnification or perspective-related distortions.

To compensate feature disparities between varying-sized objects, recently various deep-learning based counting methods using convolutional neural networks (CNNs) have been proposed. Most of them [7]–[11] are based on multi-scale features. For example, Zhang *et al.* [7] propose a multi-column network to generate features from each column and fuse these multi-scale features for crowd density estimation. Di *et al.* [11] input the pyramid of an image and adaptively fuse the multi-scale predictions for crowd counting. These multi-scale feature based methods rely on information from adjacent scales to compensate for the inability of the base CNN model to directly perceive scale variations of objects. However, at each image location there will be only one best-suited scale to the corresponding object. This implies the fusion processing will inevitably bring in interruptions from adjacent scales and may adversely affect the accuracy. In contrast, in this paper we propose to integrate the scene's geometric information into the feature manipulation and directly generate scale-aware features to mitigate the scale variation problem explicitly. Generally, the depth values at various positions can be viewed as inversely proportional to the scales of the corresponding objects. With the scale-related depth information, features can be tuned smoothly across the object scale space and the resulted scale-aware features will thus better benefit the scale-aware density estimation and crowd counting. Furthermore, considering that most existing counting datasets contain only single images, we infer depth results from a pre-trained depth prediction model [12], which makes our method more applicable.

The depth information is quite often utilized to help vision tasks [13]–[15]. The most straightforward approach is to stack the depth image as a four-channel of the input for networks. However, in experiments we found this leads to limited benefits, which has also been validated in [15] with even degraded performances. Alternatively, we propose a depth embedding module which integrates the depth information and spatially re-calibrates the magnitude at individual feature map location to generate the desired scale-aware representations. An encoding layer first encodes the depth image into the feature space. Although the encoded depth can provide geometric information, it does not help differentiate between the foreground and background areas. To specifically highlight the attentive foreground objects and avoid the distraction from the background areas, a rectify layer follows to refine the encoded depth map and generate scale-aware weights. Finally, an embedding layer applies the inferred weights to re-calibrate the magnitude of the original features at different locations for scale-aware representations.

We highlight the main contributions of this work as follows:

- Observing the effects of intra-image scale variations on density values, we propose to exploit the depth cues to directly generate scale-aware feature representations to improve the crowd density estimation.
- A depth embedding module is developed as add-ons to process the depth information and perform spatial recalibration on the features.
- With the proposed module, we implement Depth Embedded Networks (*DeemNet*) for crowd counting and demonstrate their effectiveness on four benchmark datasets.

The rest of this paper is organized as follows. Section II reviews the related work. Section III describes the proposed method. Section V presents detailed experiment results, and finally we conclude the paper in Section VII.

II. Related Work

A. Crowd Counting

Due to the limited objects' sizes and severe occlusions in modern crowd scenarios, detection-based counting methods [16]–[18] have been largely replaced by the regression-based approaches [19]–[21] to learn a mapping between the features and the crowd count. The density-based method [1] further proposes to exploit the spatial information and predict a density map whose summations across the image will report the total count. However, due to the usage of traditionally hand-crafted features, the capacity of the early counting methods have been largely limited.

Recently, the prevalence of deep learning techniques [22] has triggered a flurry of work exploring CNN-based models to improve the counting task. Zhang *et al.* [23], as the very first ones, successfully apply a seven-layer CNN for crowd counting and density estimation. Among the several factors that influence the counting accuracy, handling of the intra-image scale variations caused by the perspective-related distortions has been drawing extensive attention due to its extremely challenging situations [2]. The related methods can be mainly divided into two categories, which are summarized in Fig. 2. **The first category** is based on multi-scale features. Due to the presence of scale variations, the fusion of features from multiple scales would potentially cover the variations and thus improve the counting results. There are mainly two key components when utilizing multi-scale features: the multi-scale feature construction and their fusion. An intuitive method for multi-scale feature generation is based on the image pyramid [8], [11] that considers input images in multiple scales (Fig. 2 (a)). For example, Daniel *et al.* [8] construct a pyramid of image patches as inputs to the CNN model to emit feature maps at multiple scales. Other methods to construct multi-scale features generally based on the feature hierarchy of the deep neural networks (Fig. 2 (b)). In this situation, features are usually aggregated from sub-networks with multiple receptive fields [7], layers at an increasing depth of a CNN model [24], [25], or from the spatial pyramid pooling layer [10] for multi-scale representations. For the multi-scale feature fusion, initial methods [7], [8] treated the multi-scale features equally and simply sum/concatenate them for final prediction (Fig. 2 (c)). To further improve the fusion efficiency, later methods [9]–[11] exploit the weighted fusion mechanism and propose

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2943010, IEEE Transactions on Circuits and Systems for Video Technology

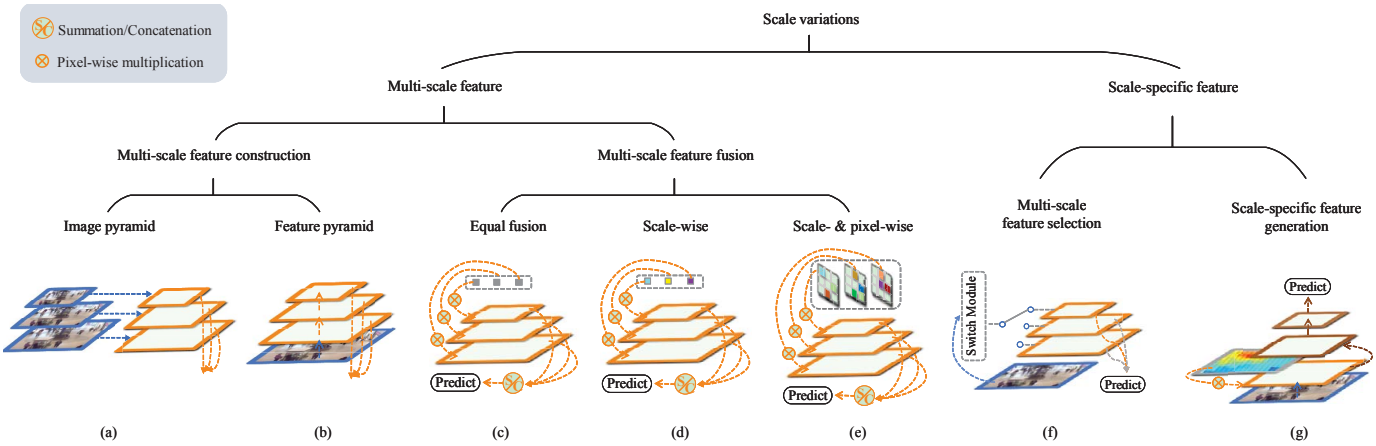


Fig. 2. Illustration of representative methods handling the scale variation problem for crowd counting.

various methods to generate adaptive weights. The learned weights can be either in the form of scalers [9] that only measure the importance of features across scales (Fig. 2 (d)), or in the form of tensors [10], [11] to measure the importance across both scales and locations of the multi-scale features (Fig. 2 (e)). For example, in [9] two additional attention models are built and learned to generate local and global scaling factors which adaptively fuse the multi-scale features for density estimation. Liu *et al.* [10] learn weighting maps to combine features emitted from the spatial pyramid layers for multi-scale representation and density estimation.

These multi-scale feature based methods rely on the information from adjacent scales to compensate for the inability of the base CNN model to directly perceive scale variations of objects. However, for a certain object there is only one best-suited scale, and thus incorporation of information from adjacent scales may bring in interruption and adversely affect the counting accuracy. As a result, **the second category** of methods to handle the scale variations aims to generate scale-specific features that could respond to the corresponding sizes specifically. A first attempt has been made in [26], where a switch module is built to control the selection of the best-suited features among several scales. Only features that are selected will be involved in final density estimation (Fig. 2 (f)). This method waivers the interruption from multi-scale fusion. However, the scale-space of objects here is discretely divided in advance with three regression models and thus the ability of the model to handle scale variations is limited by the number of the pre-defined scales. Later, Shen *et al.* [27] propose a scale-consistency loss function to drive the network to understand the scale variations and achieve counting consistency across inputs in various scales. This can be viewed as to generate scale-specific features in an implicit way. Differently, our model directly and explicitly generates scale-aware features to handle scale variations (Fig. 2 (g)). With the naturally scale-related depth maps, features can be tuned smoothly across the object scale space and the resulted scale-aware feature maps will thus better benefit the scale-aware density estimation and crowd counting.

There are also a few methods which exploit the side information of depth cues to improve crowd counting. For example, Xu *et al.* [28] use the depth of an image to guide the segmentation of the scene into a far-view region and a near-view region, and then apply different mechanisms (density-based and detection-based) to estimate counting results in the two regions. More recently, Kang *et al.* [29] propose an adaption CNN whose filter weights are derived from the central perspective value of the input patch, which disentangles the variations related to the scene perspective into model parameters. Alternatively, in this paper we directly embed the depth into the feature generation process, which models the scale variations in the input image explicitly via the generation of scale-aware representations.

B. Feature Map Attention and Scaling

The attention mechanism can perform soft selection on the relevant parts of the input to improve the representations, which has been exploited in a wide range of tasks [30], [31]. Recently, applying the attention mechanism to manipulate the feature maps in CNNs has shown to be effective. For example, Spatial Transformer Network [32] learns an in-network transformation of feature maps which can act as selectively attention to emphasize specific features. The EncNet [33] applies the channel-wise attention to emphasize or de-emphasize individual feature maps conditioned on the scene context to improve semantic segmentation. For the crowd counting, Sam *et al.* [34] also exploit top-down feedback to correct features to improve the counting performances. However, their main aim is to help differentiate between the foreground/background areas and correct the initially false positive responses, which are not specifically for the scale variation problem. In our paper, we incorporate depth cues to predict scaling factors to re-calibrate the magnitude of individual features for scale-aware representations and density estimation.

III. Approach

A. Overview

We adopt the popular encoder-decoder framework [27], [35] for crowd density estimation, where a CNN encoder

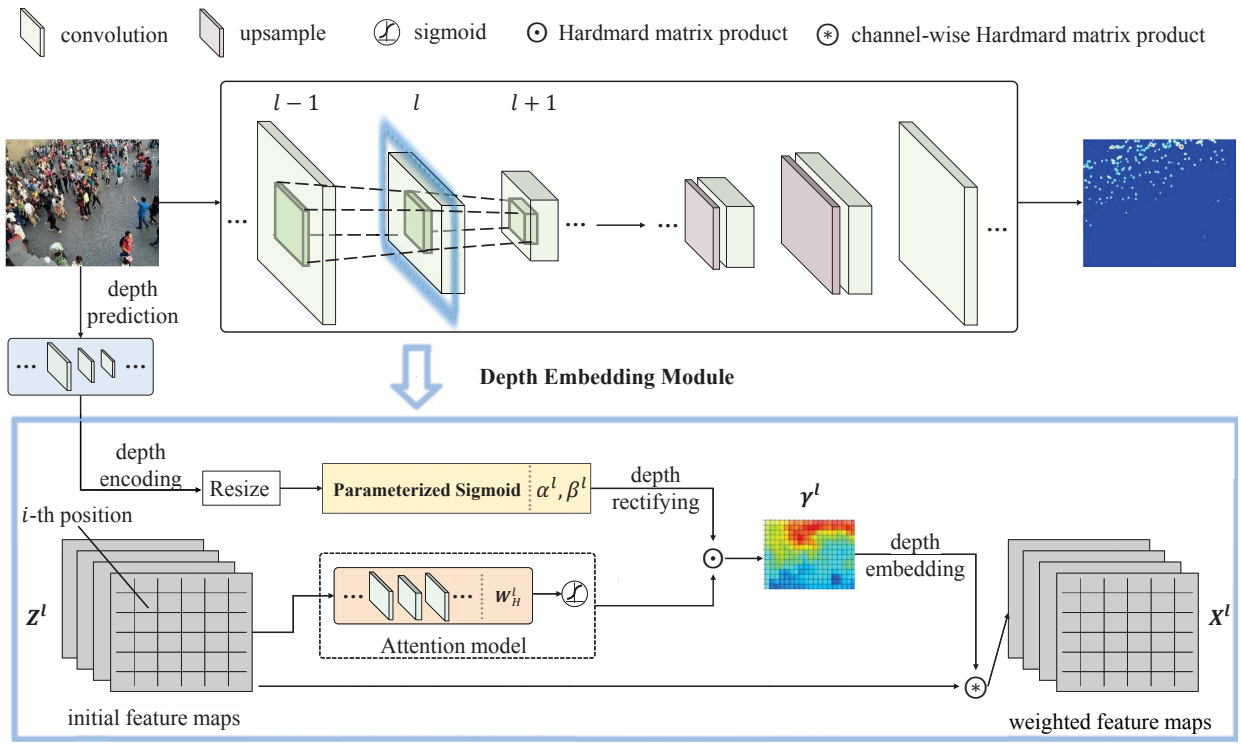


Fig. 3. Overview of the proposed Deem-CNN. For the l-the layer in the CNN encoder, initial feature maps $\mathbf{Z}^l$ is the output of the previous $(l-1)$-th layer. We build a Depth Embedding Module on top, including a depth encoding layer, a depth rectifying layer and a depth embedding layer to capture essential geometric depth cues to predict attentive scale-aware scaling weights γ^l that are conditional on the initial feature maps and the predicted depth result. The learned weights re-calibrate the magnitude of $\mathbf{Z}^l$ at each individual location, resulting in a weighted scale-aware feature map $\mathbf{X}^l$.

transforms an input image to high-level multi-layer feature maps and then a CNN decoder decodes the feature maps into a spatial density map. As illustrated in Fig. 3, our depth embedded network (DeemNet) aims to modulate the original features to embed essential geometric attribute through a depth embedding module which produces scale-aware scaling factors for individual locations on the feature maps.

Formally, suppose for the input image $\mathbf{I}$ we have its depth image $\mathbf{D}$ at hand. At the l-th layer of the encoder, the scaling factors, dubbed as scale-aware weights γ^l, is a function of $\mathbf{D}$ and the current CNN features $\mathbf{Z}^l$ at layer l. Thus, DeemNet re-calibrates current features $\mathbf{Z}^l$ using the scale-aware weights γ^l in a recurrent fashion as:

$$\begin{aligned} \mathbf{Z}^l &= \mathrm{CNN}(\mathbf{X}^{l-1}) \\ \gamma^l &= \mathcal{T}^l(\mathbf{Z}^l, \mathbf{D}) \\ \mathbf{X}^l &= f(\mathbf{Z}^l, \gamma^l), \end{aligned} \tag{1}$$

where $\mathbf{Z}^l$ is the output from previous convolution (conv) layers in the CNN model, $\mathbf{D}$ is the predicted depth image using pre-trained models (Section III-B), $\mathcal{T}^l$ denotes the transformation function that generates the scale-aware weights in the depth embedding module (Section III-C), $f(\cdot)$ denotes the weighting function that modulates CNN features with the generated weights (Section III-C), and $\mathbf{X}^l$ is the weighted feature after re-calibration. The output features will be taken as input of the next layer and proceed until the decoder which maps the scale-aware representations into scale-aware density values.

B. Depth Prediction

As an object's scale is closely related to its distance from the camera, we exploit the depth cues of an image to help model the scale variations between objects at different locations. However, currently most existing counting benchmarks contain only single RGB images. Inspired by the recent success of CNN-based depth prediction approaches, we resort to the work of Liu *et al.* [12] which learns a deep convolutional neural fields (DCNF-FCSP) model for depth prediction. This depth predictor provides an indoor version trained using NYU2 [36] dataset and an outdoor version trained using Make3D [37] dataset. In the experiments, we exploit the indoor version for the Mall [20] dataset of an indoor scene while the outdoor version for another three datasets [7], [23], [38] with outdoor scenes. We apply this pre-trained model without any changes or finetuning on the counting scenes and achieve surprisingly reasonable results. Fig. 4 visualizes the predicted depth maps for sampled crowd images. As observed, the predicted depth images can well adapt to various scene layouts and depict the distance variations at different positions to the camera imaging plane.

C. Depth Embedding Module

As depicted in Fig. 3, the depth embedding module mainly consists of three parts: depth encoding, rectifying and embedding. Each of these submodules will be described in detail in the following article.

Depth encoding Suppose for the input image $\mathbf{I}, \mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, its depth result inferred from the depth prediction

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2943010, IEEE Transactions on Circuits and Systems for Video Technology

Fig. 4. Visualization of predicted depth maps from the pre-trained depth prediction model [12]. The first row shows sample images from four crowd counting datasets [7], [20], [23], [38] and the second row visualizes their depth maps, respectively.

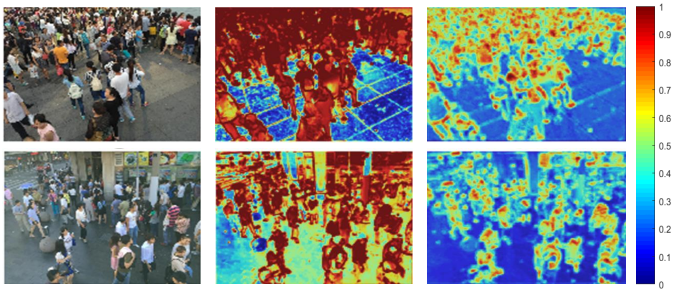


Fig. 5. Visualization of attention masks. The first column shows two sample images. The second and the third column respectively visualize the learned attention masks when the attention module is set at increasing depths of the backbone model. In all the heat maps from blue to red, the underlying value becomes larger.

model [12] is $\mathbf{D}, \mathbf{D} \in \mathbb{R}^{H \times W}$. For the depth embedding module at layer l, the depth image is first resized to match the size of feature maps $\mathbf{Z}^l$ at the corresponding layer. For scale-aware features, larger weights need to be assigned to farther, smaller-scaled objects. Considering that the desired distance information has been readily available in the depth map, we then simply employ a non-linear encoding with the parameterized sigmoid function [39] to normalize the depth values into $(0, 1)$:

$$u^l = g(\mathbf{D}) = \frac{1}{1 + e^{-\alpha^l \mathbf{D} + \beta^l}}, \tag{2}$$

where α^l and β^l are learnable parameters to tune the encoding function. This function is differentiable and hence it can be trained with the standard stochastic gradient decent (SGD) algorithms. The partial derivatives of the objective function L with respect to the parameters α^l can be written according to the chain rule as:

$$\begin{aligned} \frac{\partial L}{\partial \alpha^l} &= \frac{\partial L}{\partial u^l} \frac{\partial u^l}{\partial \alpha^l} \\ &= \sum_j \frac{\partial L}{\partial u_j^l} \frac{\partial u_j^l}{\partial \alpha^l} \\ &= \sum_j \frac{\partial L}{\partial u_j^l} \mathbf{D}_j g(\mathbf{D}_j)(1 - g(\mathbf{D}_j)) \end{aligned} \tag{3}$$

where u_j^l and $\mathbf{D}_j$ are the j-th element of u^l and $\mathbf{D}$, and the objective function L will be described in Section IV. Similarly, the partial derivatives of the objective function L with respect to β^l can be written as:

$$\frac{\partial L}{\partial \beta^l} = -\sum_j \frac{\partial L}{\partial u_j^l} g(\mathbf{D}_j)(1 - g(\mathbf{D}_j)) \tag{4}$$

Depth rectification While the depth provides information on scale variation, it is blind to the whole scene and does not specifically differentiate between foreground objects and background. With this raw depth map, features at background areas will also be inevitably re-calibrated, which is undesirable and may disrupt the originally learned feature representations. For example, the features towards the background sky at remote places (with larger depth values) will be assigned with very large weights upon a direct application of the initially encoded depth, which is irrelevant in the measurement of scale variations among target objects and also may introduce additional background noises. Towards more effective utilization of the predicted depth, we propose a rectification layer for depth refinement.

Intuitively, prior information on the potential crowd regions would be beneficial. However, at hand we only have the label of dotted annotations of pedestrians, and it is expensive to label additional crowd segments. In contrast, we introduce the spatial attention mechanism [31] to tell where the foreground objects are located in with a soft attention mask v^l for the depth embedding module at layer l. This attention mask, which de-emphasize the background areas, will act as a guide to help the model to selectively focus on the depth distinction among those targeted objects. This mask $v^l \in \mathbb{R}^{M \times N}$ can be written as a function of the feature maps $\mathbf{Z}^l \in \mathbb{R}^{M \times N \times C}$:

$$v^l = \text{sigmoid}(\Phi_s(\mathbf{Z}^l)) \tag{5}$$

where Φ_s represents a CNN-based attention model which is composed of two convolution layers with a kernel size of 3×3 (the first layer has 512 filters and the second layer has 1 filter). The attentive weights are further computed by element-wise sigmoid function on the output score map from Φ_s to highlight the most relevant regions across the whole spatial areas. In our case for crowd counting, it will learn to attend to the foreground pedestrian regions.

Fig. 5 visualizes some examples of learned attention masks. The second and the third columns respectively show the results when the input feature maps are from different layers at increasing depths of the backbone model. It can be observed that the attention masks can effectively highlight the foreground crowd areas from the background. It is also notable that with hierarchical feature representations enabled by the CNNs, it is possible to generate attention masks at different semantic levels. As observed, attention masks at increasing depths concentrate on more abstract representations, i.e., from global crowd regions to isolated head locations.

Further, the encoded depth is rectified using the attention mask to obtain the attentively scale-aware weights γ^l:

$$\gamma^l = \mathcal{T}^l(\mathbf{Z}^l, \mathbf{D}) = v^l \odot u^l \tag{6}$$

where $\odot$ denotes the Hadamard matrix product operation ($(A \odot B)_{i,j} = (A)_{i,j}(B)_{i,j}$). With the multiplicative combination,

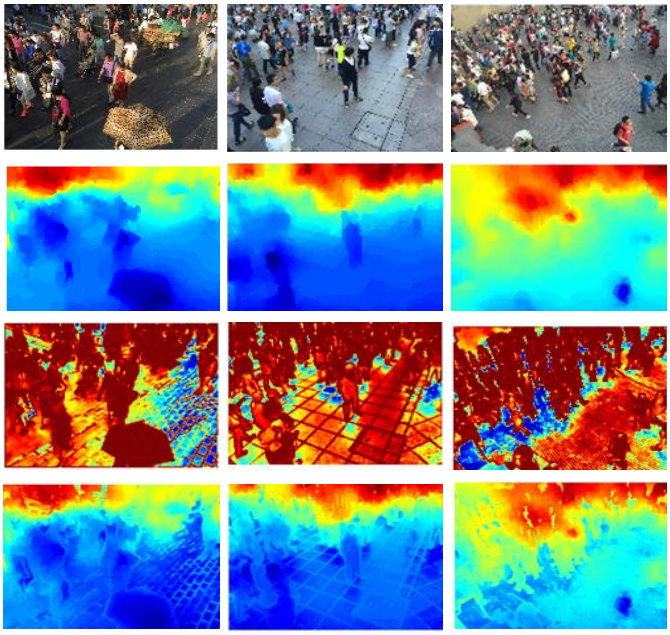


Fig. 6. Visualization of the image (first row), the depth map (second row), the attention mask (third row) and the generated attentively scale-aware weight maps after depth rectification (last row).

attention masks v^l will help rectify and suppress irrelevant signals in the background areas of the encoded depth u^l. Fig. 6 shows the effects of the depth rectification layer. As observed, after rectification the background areas are de-emphasized, whereas depth disparities among the foreground objects are still preserved and highlighted, implying the effectiveness of the rectification towards attentive scale-aware weights.

Depth embedding With the scale-aware weights, the original feature $\mathbf{Z}^l$ is tuned using a linear weighting function $f(\cdot)$ as a feedback loop. Different from the existing popular modulating strategy that aggregates features across spatial locations based on the generated weights, function $f(\cdot)$ applies element-wise multiplication. As a consequence, feature activations at different positions are re-calibrated considering both the geometry information and the semantic information at one specific position. The newly derived scale-aware feature $\mathbf{X}^l$ with highlighted scale variations among the foreground objects can be written as:

$$\mathbf{X}^l = f(\mathbf{Z}^l, \gamma^l) = \mathbf{Z}^l \circledast \gamma^l \tag{7}$$

where $\circledast$ denotes channel-wise Hardmard matrix product operation.

D. Depth Embedded Network (DeemNet)

The depth embedding module is self-contained with the same input and output dimension, and hence can be freely dropped in a standard CNN architecture to augment the representation ability, without any additional supervision or modification to the original architecture. To examine its effectiveness on backbone models with various complexity, we develop the depth embedded network (DeemNet) by integrating the proposed module into the encoder part of different backbone models. We first devise a lightweight model that has three convolution layers both in the encoder and decoder parts. This counting model is in the fully convolutional fashion and is able to accept arbitrary-sized inputs at inference, dubbed as *CFCN*. For the deeper counterpart, we exploit the most recent *CSRNet* [40] which adapts the VGG network [41] for crowd counting with dilation processing. Detailed architectures of two backbone models are shown in Table I. Besides, each convolutional layer is followed by a rectified linear unit (RELU) (omitted in the table), and is accordingly padded to keep the spatial resolution. With the two baseline CNNs, we can construct two variants of DeemNet: Deem-CFCN and Deem-CSRNet, which will be investigated in Section V.

TABLE I
DIFFERENT ENCODER-DECODER ARCHITECTURES EVALUATED IN EXPERIMENTS.

Architecture	CFCN	CSRNet
Encoder	7×7×32 conv, stride 2 7×7×64 conv, stride 2 5×5×128 conv	(3×3×64 conv)×2, stride 2 (3×3×128 conv)×2, stride 2 (3×3×256 conv)×2, stride 2 (3×3×512 conv)×2, stride 2
Decoder	5×5×64 conv 7×7×32 deconv, upsample 2 7×7×1 deconv, upsample 2	(3×3×512 conv, dilate 2)×3 3×3×256 conv, dilate 2 3×3×128 conv, dilate 2 3×3×64 conv, dilate 2 1×1×1 conv

IV. MODEL TRAINING

The DeemNet can be trained with the pixel-wise Euclidean loss: $L = \|\mathbf{Y} - \mathbf{Y}_{gt}\|^2$, where $\mathbf{Y}$ and $\mathbf{Y}_{gt}$ are the predicted and the ground-truth density map, respectively. For an image $\mathbf{I}$ with its dotted annotation set A_I, the ground-truth density map is defined as a summation of a set of 2D Gaussian functions centered at each dot, i.e., $\forall p \in \mathbf{I}, \mathbf{Y}_{gt}(p) = \sum_{\mu \in A_\mathbf{I}} \mathbb{N}(p; \mu, \Sigma)$, where $\mathbb{N}(p; \mu, \Sigma)$ denotes a normalized 2-D Gaussian kernel evaluated at p, with mean μ and isotropic covariance matrix Σ. Training proceeds in three phases: first the baseline model is optimized using objective L; then the attention model is firstly added and trained to provide better initialization; finally the complete depth embedding module is built and the whole model is trained end-to-end using L.

V. EXPERIMENTS

A. Implementation

Our system is implemented with the publicly available Matconvnet toolbox [42] with an Nvidia GTX Titan X GPU and Intel Core i7 6700 processor. We set the momentum to 0.9 and the weight decay to 0.0005. The initial learning rate is set to 10^{-5} and is divided by 10 when the validation loss plateaus. For each evaluation dataset, image patches are randomly cropped from the training images to augment the training data, and randomly flipped for data augmentation. At inference, summation of density values across the whole image will report the final counting numbers. Following the convention of most existing work [7], [23], We use the mean absolute error (MAE) and the mean square error (MSE) to evaluate and compare the counting performances. Training the whole network of Deem-CSRNet takes around 22, 35, 25 and

12.7 hours on the ShanghaiTech_B [7], worldExpo'2010 [23], UCF_CC_50 [38] and the Mall [20] dataset, respectively. At inference, it takes ∼0.1 s of the Deem-CSRNet and ∼0.05 s of the Deem-CFCN for one network forward pass to do density estimation if the depth maps are available. We will further discuss the depth efficiency and the robustness of our model in terms of the depth variation in Section VI.

B. Datasets

ShanghaiTech-B ShanghaiTech-B is a part of ShanghaiTech dataset which is proposed in [7] and is among the largest datasets captured in real outdoor scenes. It consists of 716 annotated images, which are taken by surveillance cameras from different crowd scenes. The perspective distortion in each image is pretty severe, which leads to drastic pedestrian scale variations. In our experiments, we follow the train/test splits (400 for training, 316 for testing) in the original paper [7]. 20 patches are randomly cropped from each original image for model training, each with a size of 224×224.

WorldExpo'2010 The WorldExpo'10 dataset was firstly introduced in [23]. It consists of 1132 annotated video sequences captured with 108 surveillance cameras. 3980 frames are selected and labeled with dotted annotations at the center of pedestrians' heads for evaluation of the crowd counting algorithms. Among all the images 3380 frames from 103 scenes are split as training data, and the left 600 frames from another five different scenes are held out for testing. The region of interest (ROI) and a perspective map are provided for each scene. We randomly crop 20 patches with a size of 224×224 from each training image for model learning. The ROI is used to mask the predicted density map, and only the predictions within the ROI will be considered.

UCF_CC_50 UCF_CC_50 [38] contains 50 crowd images which are crawled from the Internet. The dataset exhibits a significant variance in the counting numbers with counts varying between 94 and 4543. The limited number of training images and the drastic variability between different scenes make this dataset very challenging for the counting task. We follow the approach of other state-of-the-art methods [7], [23], [26] and use 5-fold cross-validation to validate the performance of our method on UCF_CC_50. The cropped training patch size is 224×224 in each image.

Mall The Mall dataset [20] contains 2000 frames collected in a shopping mall. As an indoor scene, the pedestrian numbers in the images of this dataset are much smaller compared to the ShanghaiTech dataset [7], with the maximum and the minimum number of people in the ROI regions being 13 and 53, respectively. However, this dataset also experiences apparent perspective distortion and illumination variations, which cause significant changes in the size and appearance of objects at different positions of the scene. Following the original experiment settings in [20], the first 800 frames are used for training, and the remaining 1200 frames are kept for testing. 12 patches are randomly cropped from each image for model training, each with a size of 160×160.

TABLE II
COMPONENT ANALYSIS ON SHANGHAITECH-B. IN EACH STAGE THE BEST MAE/MSE IS INDICATED AS BOLD AND THE SECOND BEST AS ITALIC.

Model	Stage		
	1	2	3
CFCN	13.05/21.88 (MAE/MSE)		
CFCN with RGB-D input	12.63/21.13		
A-CFCN	12.67/22.13	12.91/22.44	12.77/22.41
D-CFCN	*12.25/21.09*	*11.95/21.09*	**12.09**/*20.14*
Deem-CFCN	**11.82/19.77**	**11.86/20.48**	*12.25*/**20.05**

TABLE III
COMPONENT ANALYSIS ON MALL.

Model	Stage		
	1	2	3
CFCN	3.14/3.90 (MAE/MSE)		
CFCN with RGB-D input	2.23/2.89		
A-CFCN	3.13/3.89	2.16/2.79	2.22/2.87
D-CFCN	*2.22/2.88*	*2.14/2.72*	*2.18/2.81*
Deem-CFCN	**2.14/2.73**	**2.11/2.71**	**2.12/2.74**

C. Diagnostics Experiments

In this section, we conduct extensive experiments to analyze the effects of the proposed depth embedding module. on two datasets: ShanghaiTech-B [7] and the Mall dataset [20].

Component analysis To investigate the effects of each component in the depth embedding module, we conduct experiments with two variants of the proposed module. The first one preserves the depth encoding and embedding layers however remove the depth rectifying layer, dubbed as D-CNN. The other one only contains the attention model in the depth rectification layers and abandon the depth information, dubbed as A-CNN. To further understand the effects of the feature modulation at different positions, the depth embedding module and its variants are also applied at different stages of the base model. In particular, we denote the stage where features emitted from the n-th conv-relu-pool (or conv-relu) group as stage n. For example, for CFCN the 1st, 2nd and 3rd stage indicates the *pool1*, *pool2* and *conv3* layer, respectively. Experiment results are shown in Table II and Table III. Besides, we also include another approach which directly stacks the predicted depth map as the fourth-channel of the RGB input (RGB-D input) for comparison.

From Table II it can be observed that with merely the depth information, the D-CFCN already improves over the baseline CFCN no matter whichever stage the feature is augmented, which validates the efficacy of explicitly exploiting the predicted depth to assist the crowd counting task. When adding the depth rectification layer, the Deem-CFCN further augment the performance based on D-CFCN, implying the effectiveness to selectively highlight the attentive depth areas to avoid disruptions from the background. Besides, with only the attention model, the A-CFCN is inferior compared to Deem-CFCN. This implies that the benefits of the depth embedding module are mainly owing to the overall mechanism to encode, rectify and embed the depth information, other than the increased parameters brought by the attention block. Moreover, although exploiting the depth maps as RGB-D input

TABLE IV
DIAGNOSTIC EXPERIMENTS ON MULTI-STAGE DEPTH EMBEDDING ON SHANGHAITECH-B.

Model	Stage		
	1	1-2	1-3
CFCN	13.05/21.88 (MAE/MSE)		
Deem-CFCN	11.82/19.77	**11.34**/18.60	11.65/**18.39**
CSRNet	10.6/16.0 (MAE/MSE)		
Deem-CSRNet	8.09/**12.98**	**8.05**/13.48	8.24/14.40

TABLE V
DIAGNOSTIC EXPERIMENTS ON MULTI-STAGE DEPTH EMBEDDING ON MALL.

Model	Stage		
	1	1-2	1-3
CFCN	3.14/3.90 (MAE/MSE)		
Deem-CFCN	2.14/2.73	**2.10/2.66**	2.11/2.72
CSRNet	3.36/4.11		
Deem-CSRNet	**2.31/2.94**	2.34/2.96	2.39/3.04

also improves the performances, it is not as effective as the proposed method. Similar conclusions can also be drawn from Table III for the Mall dataset.

Multi-stage depth embedding To further investigate the effects of applying multiple depth embedding modules on the counting accuracy, we add n ($n > 1$) proposed modules at each of the first n stages in the base model. As observed in Table IV for the ShanghaiTech-B dataset, with the baseline model of CFCN, adding two modules in the first two stages is better than only using one. The performances plateau out when the third module is added, where the MSE is slightly enhanced while the MAE is degraded. For the deeper CSRNet [40], the performance stops improving when two modules being integrated and becomes even worse with the third module. It is notable that the same stage indicates layers at different depths of the CFCN and the CSRNet. Specifically, the stage 1, 2, and 3 just indicates the first, second and third convolution layer in the CFCN, whereas in CSRNet it denotes the second, fourth and sixth convolution layer, which is much deeper than CFCN. Combining the results both on CFCN and CSRNet, we found that it will be beneficial to add depth embedding modules at earlier stages to inject the scale-related information. When more modules are added at increasingly deeper layers, the higher-level representations tend to be interrupted and may adversely affect the performances. From Table V for the Mall dataset, similar observations can be drawn.

D. Comparison with State-of-the-art

The proposed method is compared with several state-of-the-art methods on four challenging benchmarks for crowd counting, as shown in Table VI, VII, VIII, IX. Since the Mall [20] dataset contains only one scene and also contains a few images, we use the Deem-CFCN with two depth embedding modules added on stage 1 and 2 to benchmark the performance on this dataset. For other datasets, Deem-CSRNet with one depth embedding module integrated on stage 1 is applied for comparison.

TABLE VI
COMPARISON RESULTS OF MAE AND MSE ON SHANGHAITECH-B.

Method	MAE	MSE
LBP + RR [43]	59.1	81.7
Crowd-CNN [23]	32.0	49.8
MCNN [7]	26.4	41.3
Switch-CNN [26]	21.6	33.4
CP-CNN [44]	20.1	30.1
DecideNet [45]	20.7	29.4
TDF-CNN [34]	20.7	32.8
ACSCP [27]	17.2	27.4
IG-CNN [46]	13.6	21.1
RReg [47]	8.7	13.5
CSRNet [40]	10.6	16.0
Deem-CSRNet	**8.09**	**12.98**

TABLE VII
COMPARISON RESULTS OF MAE ON WORLDEXPO'2010.

Method	S1	S2	S3	S4	S5	Avg
LBP + RR [43]	13.6	59.8	37.1	21.8	23.4	31.0
Crowd-CNN [23]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN [7]	3.4	20.6	12.9	13.0	8.1	11.6
Switch-CNN [26]	4.4	15.7	**10.0**	11.0	5.9	9.4
CP-CNN [44]	2.9	14.7	10.5	10.4	5.8	8.86
DecideNet [45]	2.0	13.14	8.9	17.4	4.75	9.23
CSRNet†[40]	2.32	**12.96**	14.19	10.50	3.51	8.7
Deem-CSRNet	**2.08**	14.14	12.72	**9.37**	**3.37**	**8.34**

† This is our re-implementation of the CSRNet [40]. Close average MAE (8.7) has been achieved compared to the reported MAE (8.6) in the original paper [40], however with different result on each separate scene. For comparison, we base the Deem-CSRNet on our own-implemented CSRNet.

ShanghaiTech-B As observed in Table. VI, our method outperforms the recent state-of-the-art approaches on this dataset. Especially, compared to those methods which handle scale variations mainly by employing multi-scale features [7], [23], [26], our method with the proposed depth embedding module achieve better performance, which demonstrates the efficacy to exploit the depth to model scale variations explicitly for crowd counting.

WorldExpo'2010 Table VII compares the MAE with other methods on each test scene as well as the average MAE across all the scenes. As observed, our approach outperforms previous methods with an average MAE of 8.34, demonstrating the effectiveness of the proposed method on cross-scene counting. We have noticed that with the depth embedding module, the counting errors of scene 2 increases. Based on our analysis, in this scene, the ROI regions are almost directly under the surveillance camera, where the perspective distortion and the scale variation are not the dominant factors influencing the counting accuracy. Thus, the ability of the proposed method that mainly tackles the scale variations is limited in this scene.

UCF_CC_50 Images in this datasets exhibit a large diversities of crowd densities, which make it very challenging for a method to successfully adapt to such variations without any prior information. As observed in Table VIII, Deem-CSRNet improves over the baseline model and achieves the best MAE compared to other state-of-the-art methods, implying the effectiveness of the proposed method on extreme dense scenes. The CP-CNN [44] achieves the best MSE. This method

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2943010, IEEE Transactions on Circuits and Systems for Video Technology

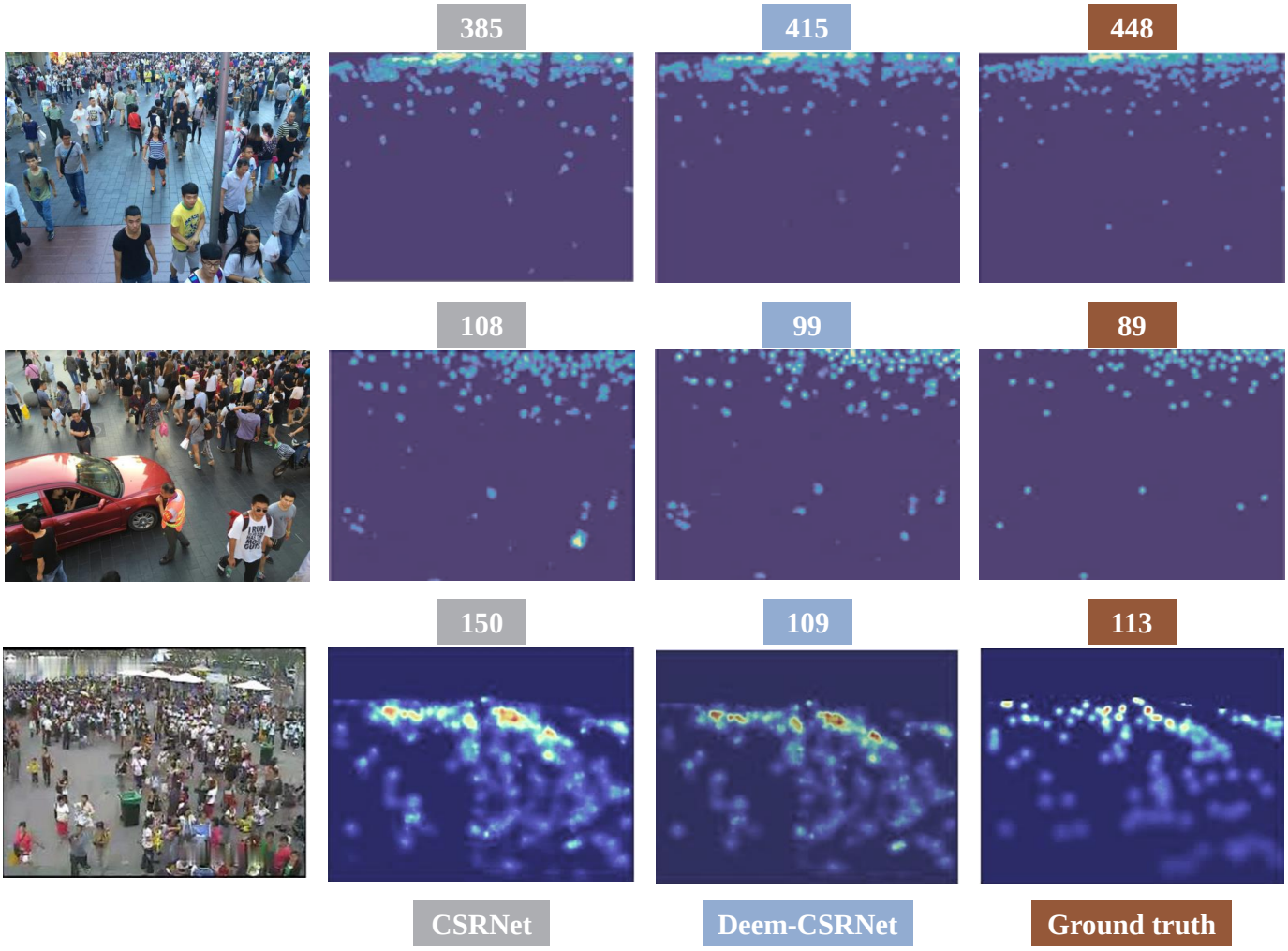


Fig. 7. Qualitative visualization. From the first to the last column are: the images, estimated density maps of the baseline model CSRNet, estimated density maps of model Deem-CSRNet with the depth embedding module, and the ground truth density maps. Crowd counts are labeled on the top for illustration.

TABLE VIII
COMPARISON RESULTS OF MAE AND MSE ON UCF_CC_50.

Method	MAE	MSE
Lempitsky *et al.* [1]	493.4	487.1
Idrees *et al.* [38]	419.5	541.6
Crowd-CNN [23]	467.0	498.5
MCNN [7]	377.6	509.1
MoCNN [48]	361.7	493.3
Hydra2s [8]	333.7	425.3
Switch-CNN [26]	318.1	439.2
CP-CNN [44]	295.8	**320.9**
ACSCP [27]	291.0	404.6
Mohammad *et al.* [9]	271.60	391.00
PACNN [49]	267.9	357.8
CSRNet [40]	266.1	397.5
Deem-CSRNet	**253.4**	364.4

TABLE IX
COMPARISON RESULTS OF MAE AND MSE ON MALL.

Method	MAE	MSE
Ridge Regression [43]	3.59	19.0
MORR [20]	3.15	15.7
Count Forest [50]	4.40	2.40
Weighted VLAD [21]	2.41	9.12
Exemplary Density [51]	1.82	2.74
Boosting CNN [52]	2.01	N/A
MoCNN [48]	2.75	13.4
DecideNet [45]	**1.52**	**1.90**
CFCN	3.14	3.90
Deem-CFCN	2.10	2.66

incorporates prior information on the density levels of an input image and its sub-patches, which makes it especially beneficial for datasets with large density diversities.

Mall As observed in Table IX, with two depth embedding modules injected in the baseline model, the performance improves and is comparable with other methods, which demonstrates the effectiveness and robustness of the proposed approach on small datasets with fewer people.

Fig. 7 qualitatively visualizes and compares the density maps and estimated counts of models with (Deem-CSRNet) and without (CSRNet) the depth embedding module. As observed, with the proposed module the estimated density map become more close to the ground truth, and also the estimated counts become more accurate. For example, for the second

sample image after depth embedding the response in lower-right regions are weakened and become more close to the ground truth, implying the ability of the Deem-CSRNet to perceive and adapt to object scales.

VI. Depth Efficiency and Model Robustness

In this paper, we input the RGB crowd images into the depth prediction model and exploit the predicted depth in our counting algorithms. However, the depth estimation, as a dense prediction task, requires a large number of computational resources. Although the depth predictor [12] has been sped up to reduce the computational burden, it is still rather time-consuming. Take the shanghaiTech-B [7] for example, it takes 4.2 s to estimate a depth map for an image in a resolution of 1024×768.

At the training stage, since the depth prediction model is fixed and does not involve in the learning process, we can collect the depth maps off-line to avoid on-line depth estimation and save the training time. However, at inference this time cannot be neglected anymore. Considering that in most real-world applications, the surveillance cameras are stationary for a certain scene with a fixed viewpoint, the depth relationships at different positions can be hence regarded unchanged across different frames. In this situation, the depth map used in the counting model can be predicted for just once when the system starts up or predicted every a few frames, thus avoiding a large proportion of time spent for depth inference.

Beyond this engineering solution, we also study another method to improve the efficiency to facilitate the utilization of predicted depth information. We find that the time for depth prediction is significantly influenced by the image size fed into the depth prediction model [12]. Table X shows the depth inference time and the corresponding counting results with downsampled depth results on the ShanghaiTech-B [7] dataset. It can be observed that generally the larger the image size, the more time it takes to do the depth estimation. Intuitively, we consider use downsampled crowd images for depth prediction to save time. The output depth map, which is also downsampled, will be bilinearly upsampled to recover their full resolution as the original crowd image before they are accepted by the depth embedding module. As observed from Table X, a suitable choice for the ShanghaiTech-B dataset at inference would be the usage of a downsampled crowd image with a downsampling factor k between 0.4-0.6. This will maintain most of the performance but take much less time for the depth estimation compared to using the full-sized images. Thus, in practical applications, when higher-resolution inputs hinder the test efficiency, the time for depth inference can also be saved with resized images. The suitable sizes may dependent on each dataset and need to be chosen accordingly. Furthermore, as observed from Table X, the proposed method takes effects and improves over the baseline even with downsampled depth maps at a wide range of sizes, which to some extent can demonstrate the robustness of our system to the depth variations.

TABLE X
Inference Time and Counting Performances Using Downsampled Depth Maps at Multiple Downsampling Factors. Assuming the Original Image Size is $M \times N$, with Downsampling Factor k, the Resized Image will be $(M \cdot k) \times (N \cdot k)$.

Downsampling factor k	Time (s)	Deem-CFCN	
		MAE	MSE
Baseline (CFCN)	-	13.05	21.88
1 (default)	4.2	**11.34**	*18.60*
0.8	2	*11.35*	18.72
0.6	0.8	11.56	**18.50**
0.4	0.3	12.02	19.47
0.2	0.07	13.31	20.97

VII. Conclusion

To handle the intra-image scale variations of pedestrians in the visual crowd counting task, we propose a novel depth embedding module to improve the representation capacity on scale variations of a network by dynamically spatial-wise feature recalibration with rectified depth cues. The proposed depth embedding module is fully differentiable and compatible with existing CNN-based approaches. Extensive experiments demonstrate the effectiveness of the depth embedded networks (Deem-CNN) which achieve state-of-the-art performance on multiple datasets. The contribution of Deem-CNN is not only a more powerful counting model but also are some insights into the limitations of plain CNN architectures in perceiving and modeling the scale variations to generate scale-aware features, which we hope may be useful for other tasks requiring awareness to the scene geometrics. Finally, we expect the strategy of depth embedding can be beneficial to vehicle counting and other general object counting tasks.

References

[1] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, 2010, pp. 1324–1332.

[2] V. A. Sindagi and V. M. Patel, "A survey of recent advances in cnn-based single image crowd counting and density estimation," *Pattern Recognition Letters*, 2017.

[3] C. Zhang, K. Kang, H. Li, X. Wang, R. Xie, and X. Yang, "Data-driven crowd understanding: A baseline for a large-scale crowd dataset," *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1048–1061, Jun 2016.

[4] C. C. Loy, K. Chen, S. Gong, and T. Xiang, "Crowd counting and profiling: Methodology and evaluation," in *Modeling, Simulation and Visual Analysis of Crowds*, ser. The International Series in Video Computing. Springer New York, 2013, vol. 11, ch. chapter 14, pp. 347–382.

[5] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Computer Vision and Pattern Recognition, 2008. IEEE Conference on*. IEEE, 2008, pp. 1–7.

[6] L. Fiaschi, U. Köthe, R. Nair, and F. A. Hamprecht, "Learning to count with regression forest and structured labels," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 2685–2688.

[7] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 589–597.

[8] D. Onoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *European Conference on Computer Vision*. Springer, 2016, pp. 615–629.

[9] M. Hossain, M. Hosseinzadeh, O. Chanda, and Y. Wang, "Crowd counting using scale-aware attention networks." IEEE, Jan 2019, pp. 1280–1288.

[10] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5099–5108.

[11] D. Kang and A. B. Chan, "Crowd counting by adaptively fusing predictions from an image pyramid," in *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, 2018, p. 89.

[12] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2024–2039, 2016.

[13] B. Ni, Y. Pei, P. Moulin, and S. Yan, "Multilevel depth and image fusion for human activity detection," *IEEE transactions on cybernetics*, vol. 43, no. 5, pp. 1383–1394, 2013.

[14] Y. Cao, C. Shen, and H. T. Shen, "Exploiting depth from single monocular images for object detection and semantic segmentation," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 836–846, 2017.

[15] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture," in *Proc. ACCV*, vol. 2, 2016.

[16] I. S. Topkaya, H. Erdogan, and F. Porikli, "Counting people by clustering person detector outputs," in *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*. IEEE, 2014, pp. 313–318.

[17] C. Gao, P. Li, Y. Zhang, J. Liu, and L. Wang, "People counting based on head detection combining adaboost and cnn in crowded surveillance environment," *Neurocomputing*, vol. 208, pp. 108–116, Oct 2016.

[18] V. B. Subburaman, A. Descamps, and C. Carincotte, "Counting people in the crowd using a generic head detector." IEEE, Sep 2012, pp. 470–475.

[19] D. Kong, D. Gray, and H. Tao, "A viewpoint invariant approach for crowd counting." IEEE, 2006, pp. 1187–1190.

[20] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting." in *BMVC*, vol. 1, no. 2, 2012, p. 3.

[21] B. Sheng, C. Shen, G. Lin, J. Li, W. Yang, and C. Sun, "Crowd counting via weighted vlad on dense attribute feature maps," *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[23] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 833–841.

[24] V. Ranjan, H. Le, and M. Hoai, "Iterative crowd counting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 270–285.

[25] L. Zhang, M. Shi, and Q. Chen, "Crowd counting via scale-adaptive convolutional neural network." IEEE, Mar 2018, pp. 1113–1121.

[26] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting." IEEE, Jul 2017, pp. 4031–4039.

[27] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang, "Crowd counting via adversarial cross-scale consistency pursuit," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5245–5254.

[28] M. Xu, Z. Ge, X. Jiang, G. Cui, P. Lv, and B. Zhou, "Depth information guided crowd counting for complex crowd scenes," *arXiv preprint arXiv:1803.02256*, 2018.

[29] D. Kang, D. Dhar, and A. Chan, "Incorporating side information by adaptive convolution," in *Advances in Neural Information Processing Systems*, 2017, pp. 3868–3878.

[30] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation." Association for Computational Linguistics, 2015, pp. 1412–1421. [Online]. Available: http://aclweb.org/anthology/D15-1166

[31] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[32] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[33] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7151–7160.

[34] D. B. Sam and R. V. Babu, "Top-down feedback for crowd counting convolutional neural network," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[35] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.

[36] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*. Springer, Berlin, Heidelberg, 2012, pp. 746–760.

[37] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 824–840, 2009.

[38] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images." IEEE, Jun 2013, pp. 2547–2554.

[39] C. Zhang and P. C. Woodland, "Parameterised sigmoid and relu hidden activation functions for dnn acoustic modelling," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[40] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1091–1100.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[42] A. Vedaldi and K. Lenc, "Matconvnet, convolutional neural networks for matlab." ACM Press, 2015, pp. 689–692.

[43] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," 1998.

[44] V. A. Sindagi and V. M. Patel, "Generating high-quality crowd density maps using contextual pyramid cnns," in *IEEE International Conference on Computer Vision*, 2017.

[45] J. Liu, C. Gao, D. Meng, and A. G. Hauptmann, "Decidenet: Counting varying density crowds through attention guided detection and density estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5197–5206.

[46] D. B. Sam, N. N. Sajjan, R. V. Babu, and M. Srinivasan, "Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn," *arXiv preprint arXiv:1807.09993*, 2018.

[47] J. Wan, W. Luo, B. Wu, A. B. Chan, and W. Liu, "Residual regression with semantic prior for crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4036–4045.

[48] S. Kumagai, K. Hotta, and T. Kurita, "Mixture of counting cnns," *Machine Vision and Applications*, Jul 2018.

[49] M. Shi, Z. Yang, C. Xu, and Q. Chen, "Revisiting perspective information for efficient crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7279–7288.

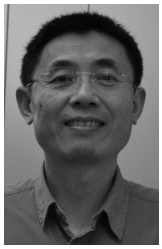
[50] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, "Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3253–3261.

[51] Y. Wang and Y. Zou, "Fast visual object counting via example-based density estimation." IEEE, Sep 2016, pp. 3653–3657.

[52] E. Walach and L. Wolf, "Learning to count with cnn boosting," in *European Conference on Computer Vision*. Springer, 2016, pp. 660–676.

Muming Zhao received the BEng degree in electronic and information Engineering from Xidian University, China, in 2013. She is currently working towards the Ph.D. degree in Shanghai Jiao Tong University (SJTU), and also as a joint-degree Ph.D. student in University of Technology Sydney (UTS), Sydney. Her research interests are in the areas of computer vision, crowd counting and deep learning.

Chongyang Zhang received the B.S. and M.S. degree from Air Force Engineering University, Xi'an, China, in 1997 and 2000, respectively, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2008. He is currently an Associate Professor of the Institute of Image Communication and Information Processing, Department of Electronic Engineering, Shanghai Jiao Tong University. His research interests are in the area of machine learning and computer vision, especially on object detection, crowd counting, action recognition, and event detection. He has published over 50 international journal or conference papers on these topics. He has served on various international conferences or journals as a reviewer, TPC member, or session chair. In 2015, he won the second prize of national science and technology progress of China.



Jian Zhang received his Ph.D. in electrical engineering from the University of New South Wales (UNSW), Sydney, Australia, in 1999. He is currently an Associate Professor with the Global Big Data Technologies Centre, School of Electrical and Data Engineering, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney. He is the author or co-author of more than 180 paper publications, book chapters, and seven issued US and China patents. His current research interests include social multimedia signal processing, large scale image and video content analytics, retrieval and mining, 3D based computer vision and intelligent video surveillance systems. He was the General Co-Chair of the International Conference on Multimedia and Expo in 2012 and Technical Program Co-Chair of IEEE Visual Communications and Image Processing 2014. Currently, he is an Associated Editors for the IEEE Transactions on Multimedia and was an Associate Editor for the IEEE Transactions on Circuits and Systems for Video Technology (2006-2015). He is a technical program co-chair of the International Conference on Multimedia and Expo in 2020 in London.

Fatih Porikli received his Ph.D. degree from New York University (NYU) in 2002. He is an IEEE Fellow and a Professor in the Research School of Engineering, Australian National University (ANU). He is also serving as the Technical Vice President at Futurewei Device & Hardware in San Diego. He led the Computer Vision Research Group Leader at NICTA, Australia and managed projects as the Distinguished Research Scientist at Mitsubishi Electric Research Laboratories, Cambridge. He developed satellite imaging solutions at HRL, Malibu CA, and 3D display systems at AT&T Research Laboratories, Middletown, NJ. His research interests include computer vision, deep learning, manifold learning, online learning, and image enhancement with commercial applications in mobile phones, AR/VR, autonomous vehicles, video surveillance, defense, and medical systems. He received the R&D 100 Scientist of the Year Award in 2006, won six best paper awards and recognized with six professional prizes at his industrial appointments. He authored more than 250 publications, co-edited two books, and invented 80 US patents. He served as the General Chair and Technical Program Chair of many IEEE conferences and an Associate Editor of premier IEEE and Springer journals for the past 15 years.

Bingbing Ni received his Ph.D. from National University of Singapore (NUS), Singapore in 2011. He is currently a Professor in Shanghai Jiao Tong University. His research interests are in the areas of computer vision, machine learning and multimedia. He received the Best Paper Award from PCM'11 and the Best Student Paper Award from PREMIA'08.

Wenjun Zhang received his B.S., M.S. and Ph.D. degrees in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 1984, 1987 and 1989, respectively. After three years' working as an engineer at Philips in Nuremberg, Germany, he went back to his Alma Mater in 1993 and became a full professor of Electronic Engineering in 1995. He was one of the main contributors of the Chinese DTTB Standard (DTMB) issued in 2006. He holds 142 patents and published 110 papers in international journals and conferences. He is the Chief Scientist of the Chinese Digital TV Engineering Research Centre (NERC-DTV), an industry/government consortium in DTV technology research and standardization, and the director of Cooperative MediaNet Innovation Center (CMIC), an excellence research cluster affirmed by the Chinese Government. His main research interests include video coding and wireless transmission, multimedia semantic analysis and broadcast/broadband network convergence.