Cross-scene Crowd Counting via Deep Convolutional Neural Networks

Cong Zhang[1,2] Hongsheng Li[2,3] Xiaogang Wang[2] Xiaokang Yang[1]
[1]Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University
[2]Department of Electronic Engineering, The Chinese University of Hong Kong
[3]School of Electronic Engineering, University of Electronic Science and Technology of China
{zhangcong0929,lihongsheng}@gmail.com xgwang@ee.cuhk.edu.hk xkyang@sjtu.edu.cn

Abstract

Cross-scene crowd counting is a challenging task where no laborious data annotation is required for counting people in new target surveillance crowd scenes unseen in the training set. The performance of most existing crowd counting methods drops significantly when they are applied to an unseen scene. To address this problem, we propose a deep convolutional neural network (CNN) for crowd counting, and it is trained alternatively with two related learning objectives, crowd density and crowd count. This proposed switchable learning approach is able to obtain better local optimum for both objectives. To handle an unseen target crowd scene, we present a data-driven method to fine-tune the trained CNN model for the target scene. A new dataset including 108 crowd scenes with nearly 200,000 head annotations is introduced to better evaluate the accuracy of cross-scene crowd counting methods. Extensive experiments on the proposed and another two existing datasets demonstrate the effectiveness and reliability of our approach.

1. Introduction

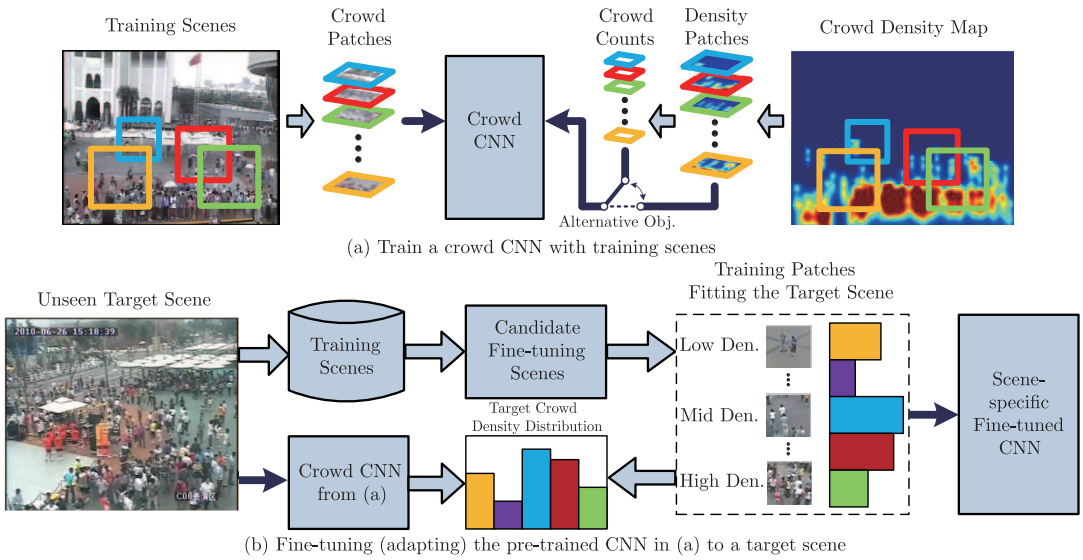
Counting crowd pedestrians in videos draws a lot of attention because of its intense demands in video surveillance, and it is especially important for metropolis security. Crowd counting is a challenging task due to severe occlusions, scene perspective distortions and diverse crowd distributions. Since pedestrian detection and tracking has difficulty when being used in crowd scenes, most state-of-the-art methods [6, 4, 5, 17] are regression based and the goal is to learn a mapping between low-level features and crowd counts. However, these works are scene-specific, i.e., a crowd counting model learned for a particular scene can only be applied to the same scene. Given an unseen scene or a changed scene layout, the model has to be re-trained with new annotations. There are few works focusing on cross-scene crowd counting, though it is important to actual applications.

In this paper, we propose a framework for cross-scene crowd counting. No extra annotations are needed for a new target scene. Our goal is to learn a mapping from images to crowd counts, and then to use the mapping in unseen target scenes for cross-scene crowd counting. To achieve this goal, we need to overcome the following challenges. 1) Develop effective features to describe crowd. Previous works used general hand-crafted features, which have low representation capability for crowd. New descriptors specially designed or learned for crowd scenes are needed. 2) Different scenes have different perspective distortions, crowd distributions and lighting conditions. Without additional training data, the model trained in one specific scene has difficulty being used for other scenes. 3) For most recent works, foreground segmentation is indispensable for crowd counting. But crowd segmentation is a challenging problem and can not be accurately obtained in most crowded scenes. The scene may also have stationary crowd without movement. 4) Existing crowd counting datasets are not sufficient to support and evaluate cross-scene counting research. The largest one [8] only contains 50 static images from different crowd scenes collected from Flickr. The widely used UCSD dataset [4] and the Mall dataset [6] only consist of video clips collected from one or two scenes.

Considering these challenges, we propose a Convolutional Neural Network (CNN) based framework for cross-scene crowd counting. After a CNN is trained with a fixed dataset, a data-driven method is introduced to fine-tune (adapt) the learned CNN to an unseen target scene, where training samples similar to the target scene are retrieved from the training scenes for fine-tuning. Figure 1 illustrates the overall framework of our proposed method. Our cross-scene crowd density estimation and counting framework has following advantages:

1. Our CNN model is trained for crowd scenes by a switchable learning process with two learning objectives, crowd density maps and crowd counts. The two different but related objectives can alternatively assist each other to

978-1-4673-6964-0/15/$31.00 ©2015 IEEE



(a) Train a crowd CNN with training scenes

(b) Fine-tuning (adapting) the pre-trained CNN in (a) to a target scene

Figure 1. Illustration of our proposed cross-scene crowd counting method.

obtain better local optima. Our CNN model learns crowd-specific features, which are more effective and robust than handcrafted features.

2. The target scenes require no extra labels in our framework for cross-scene counting. The pre-trained CNN model is fine-tuned for each target scene to overcome the domain gap between different scenes. The fine-tuned model is specifically adapted to the new target scene.

3. The framework does not rely on foreground segmentation results because only appearance information is considered in our method. No matter whether the crowd is moving or not, the crowd texture would be captured by the CNN model and can obtain a reasonable counting result.

4. We also introduce a new dataset [1] for evaluating cross-scene crowd counting methods. To the best of our knowledge, this is the largest dataset for evaluating crowd counting algorithms.

2. Related work

Counting by global regression. Many works have been proposed to count the pedestrians by detection [29, 14, 27] or trajectory-clustering [3, 21]. But for the crowd counting problem, these methods are limited by severe occlusions between people. A number of methods [1, 6, 4, 5, 11] tried to predict global counts by using regressors trained with low-level features. These approaches are more suitable for crowded environments and is computationally more efficient. Loy *et al.* [17] introduced semi-supervised regression and data transferring methods to reduce the amount of training data needed, but it still needs some labels from the target crowd scene. Idrees *et al.* [8] estimated the number of individuals in dense crowds based on multi-source information from images but not surveillance videos.

Counting by density estimation estimation. Counting by global regression ignores spatial information of pedestrians. Lempitsky *et al.* [12] introduced an object counting method through pixel-level object density map regression. Following this work, Fiaschi *et al.* [7] used random forest to regress the object density and improve training efficiency. Besides considering spatial information, another advantage of density regression based methods is that they are able to estimate object counts in any region of an image. Taking this advantage, an interactive object counting system was introduced in [2], which visualized region counts to help users to determine the relevance feedback efficiently. And Rodrigueze [22] made use of density map estimation to improve the head detection results. These methods are scene-specific and not applicable to cross-scene counting.

Deep learning. Many works introduced deep learning into various surveillance applications, such as person re-identification [13], pedestrian detection [30, 31, 20], tracking [28], crowd behavior analysis [9] and crowd segmentation [10]. Their success benefits from discriminative power of deep models. Sermanet *et al.* [25] showed that the features extracted from deep models are more effective than hand-crafted feature for many applications. To the best of our knowledge, however, deep models have not yet been explored for crowd counting.

Data-driven approaches for scene labeling. As many large-scale and well-labeled datasets published, nonparametric, data-driven approaches [15, 26, 23] are proposed. Such approaches can be scaled up easily because they do not require training. They transfer the labels from the training images to the test image by retrieving the most sim-

[1] http://www.ee.cuhk.edu.hk/∼xgwang/expo.html

ilar training images and match them with the test image. Liu *et al.* [15] proposed a nonparametric image parsing method looking for a dense deformation field between images. Inspired by the data-driven scene labeling methods, for a unseen target scene, we retrieve similar scenes and crowd patches from the training scenes. However, instead of directly transferring labels to the target scene like existing methods, we propose to use the training samples that fits the estimated crowd density distribution to fine-tune (adapt) the pre-trained CNN model to the target scene.

3. Method

3.1. Normalized crowd density map for training

The main objective for our crowd CNN model is to learn a mapping $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{D}$, where $\mathcal{X}$ is the set of low-level features extracted from training images and $\mathcal{D}$ is the crowd density map of the image. Assuming that the position of each pedestrian is labeled, the density map is created based on pedestrians' spatial location, human body shape and perspective distortion of images. Patches randomly selected from the training images are treated as training samples, and the density maps of corresponding patches are treated as the ground truth for the crowd CNN model. As an auxiliary objective, the total crowd number in a selected training patch is calculated through integration over the density map. Note that the total number will be a decimal, but not an integer.

Many works followed [12] and defined the density map regression ground truth as a sum of Gaussian kernels centered on the locations of objects. This kind of density maps is suitable for characterizing the density distribution of circle-like objects such as cells and bacteria. However, this assumption may fail when it comes to the pedestrian crowd, where cameras are generally not in a bird-view. An example of pedestrians in an ordinary surveillance camera is shown in Figure 2. It has three visible characteristics: 1) pedestrian images in the surveillance videos have different scales due to perspective distortion; 2) the shapes of pedestrians are more similar to ellipses than circles; 3) due to severe occlusions, heads and shoulders are the main cues to judge whether there exists a pedestrian at each position. The body parts of pedestrians are not reliable for human annotation. Taking these characteristics into account, the crowd density map is created by the combination of several distributions with perspective normalization.

Perspective normalization is necessary to estimate the pedestrian scales. Inspired by [4], for each scene, we randomly select several adult pedestrians and label them from head to toe. Assuming that the mean height of adults is 175 cm, the perspective map M can be approximated through a linear regression as shown in Figure 2 (a). The pixel value in the perspective map $M(p)$ denotes that the number of pixels in the image representing one meter at that location

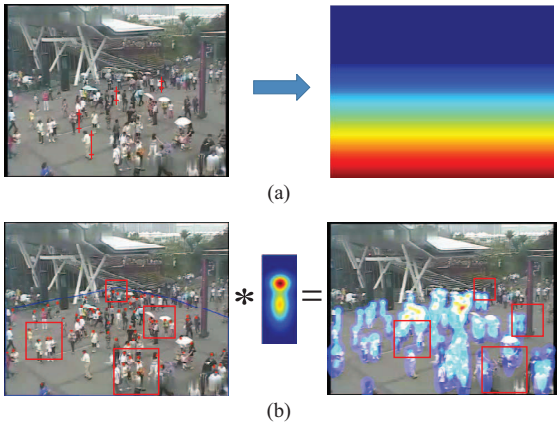


Figure 2. (a) Estimating the perspective map. Hot color indicates a high value in the perspective map. (b) The crowd density map and the red box show some training patch randomly cropped from image and density map. The patches cover the same actual area. The ones in the further away regions are smaller and the ones in the closer regions are larger.

in the actual scene. After we obtain the perspective map and the center positions of pedestrian head P_h in the region of interest (ROI), we create the crowd density map is created as:

$$\mathcal{D}_i(p) = \sum_{P \in \mathbf{P}_i} \frac{1}{\|\mathrm{Z}\|} (\mathcal{N}_h(p; P_h, \sigma_h) + \mathcal{N}_b(p; P_b, \Sigma)) \quad (1)$$

The crowd density distribution kernel contains two terms, a normalized 2D Gaussian kernel $\mathcal{N}_h$ as a head part and a bivariate normal distribution $\mathcal{N}_b$ as a body part. Here P_b is the position of the pedestrian body, estimated by the head position and the perspective value. To best represent the pedestrian contour, we set the variance $\sigma_h = 0.2M(p)$ for the term $\mathcal{N}_h$, and $\sigma_x = 0.2M(p)$, $\sigma_y = 0.5M(p)$ for the term $\mathcal{N}_b$. To ensure that the integration of all density values in a density map equals to the total crowd number in the original image, the whole distribution is normalized by Z. The crowd density distribution kernel and created density map are visualized in Figure 2 (b).

3.2. Crowd CNN model

An overview of our crowd CNN model with switchable objectives is shown in Figure 3. The input is the image patches cropped from training images. In order to obtain pedestrians at similar scales, the size of each patch at different locations is chosen according to the perspective value of its center pixel. Here we constrain each patch to cover a 3-meter by 3-meter square in the actual scene as shown in Figure 2. Then the patches are warped to 72 pixels by 72 pixels as the input of the Crowd CNN model. Our Crowd CNN model contains 3 convolution layers (con1-conv3) and

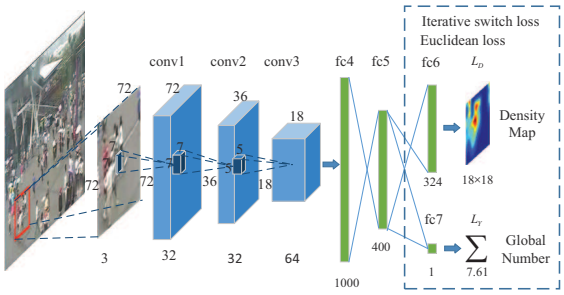


Figure 3. The structure of the crowd convolutional neural network. At the loss layer, a density map loss and a global count loss is minimized alternatively.

three fully connected layers (fc4, fc5 and fc6 or fc7). Conv1 has 32 $7\times7\times3$ filters, conv2 has 32 $7\times7\times32$ filters and the last convolution layer has 64 $5\times5\times32$ filters. Max pooling layers with a 2×2 kernel size are used after conv1 and conv2. Rectified linear unit (ReLU), which is not shown in Figure 3, is the activation function applied after every convolutional layer and fully connected layer.

We introduce an iterative switching process in our deep crowd model to alternatively optimize the density map estimation task and the count estimation task. The main task for the crowd CNN model is to estimate the crowd density map of the input patch. Because two pooling layers exist in the CNN model, the output density map is down-sampled to 18×18. Therefore, the ground truth density map is also down-sampled to 18×18. Since the density map contains rich and abundant local and detailed information, the CNN model can benefit from learning to predict density map and can obtain a better representation of crowd patches. The total count regression of the input patch is treated as the secondary task, which is calculated by integrating the density map patch. Two tasks alternatively assist each other and obtain a better solution. The two loss functions are defined as:

$$L_D(\Theta) = \frac{1}{N}\sum_{i}^{N}\|F_d(X_i;\Theta) - D_i\|^2, \tag{2}$$

$$L_Y(\Theta) = \frac{1}{N}\sum_{i}^{N}\|F_y(X_i;\Theta) - Y_i\|^2, \tag{3}$$

where Θ is the set of parameters of the CNN model and N is the number of training samples. L_D is the loss between estimated density map $F_d(X_i;\Theta)$ (the output of fc6) and the ground truth density map D_i. Similarly, L_Y is the loss between the estimated crowd number $F_y(X_i;\Theta)$ (the output of fc7) and the ground truth number Y_i. Euclidean distance is adopted in these two objective losses. The loss is minimized using mini-batch gradient descent and back-propagation.

The switchable training procedure is summarized in Algorithm 1. We set L_D as the first objective loss to minimize, since the density map can introduce more spatial information to the CNN model. Density map estimation requires the model to learn a general representation for crowd. Then after the first objective converges, the model switches to minimize the objective of global count regression. Count regression is an easier task and its learning converges faster than the task of density map regression. Note that the two objective losses should be normalized to similar scales, otherwise the objective with the larger scale would be dominant in the training process. In the experiment, we set the scale weight of density loss to 10, and the scale weight of count loss to 1. The training loss converged after about 6 switch iterations. Our proposed switching learning appoarch can achieve better performance than the widely used multi-task learning approach (see experiments in the Section 5).

Algorithm 1: Training with iterative switching losses

Input: Training set: size-normalized patches with their counts and density maps from the whole training data

Output: Parameters Θ for crowd CNN model

```
set L_D as the first objective;
for t = 1 to T do
    BP to learn Θ, until the validation loss drop rate
    ΔL is less than the threshold ε
    Switch the objective loss function
end
```

4. Nonparametric fine-tuning for target scene

The crowd CNN model is pre-trained based on all training scene data through our proposed switchable learning process. However, each query crowd scene has its unique scene properties, such as different view angles, scales and different density distributions. These properties significantly change the appearance of crowd patches and affect the performance of the crowd CNN model. In order to bridge the distribution gap between the training and test scenes, we design a nonparametric fine-tuning scheme to adapt our pre-trained CNN model to unseen target scenes. Given a target video from the unseen scenes, samples with similar properties from the training scenes are retrieved and added to training data to fine-tune the crowd CNN model. The retrieval task consists of two steps, candidate scenes retrieval and local patch retrieval.

4.1. Candidate scene retrieval

The view angle and the scale of a scene are the main factors affecting the appearance of crowd. The perspective map can indicate both the view angle and the scale as shown in Figure 2 (a). To overcome the scale gap be-

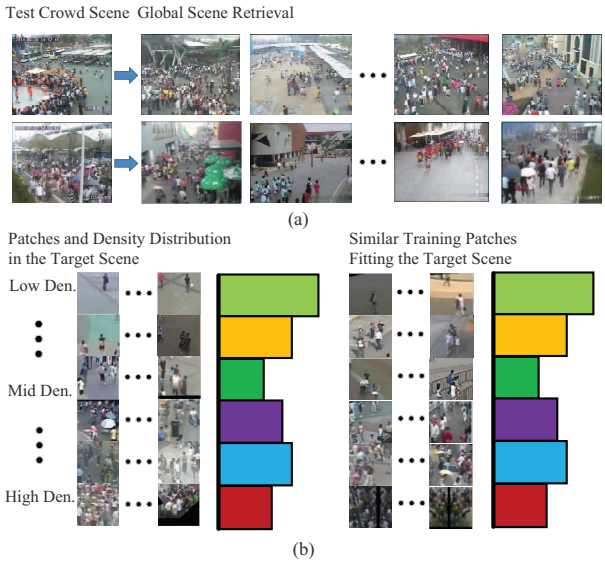


Figure 4. Illustration of retrieving local patches similar to those in the test scene to fine-tune the crowd CNN model. (a) Retrieving candidate scenes by matching perspective maps of the training scenes and the test scene. (b) Local patches similar to those in the test scene are retrieved from the candidate scenes. The color bars indicate the density distributions of patches from the test scene, and those patches selected from the train scenes

tween different scenes, each input patch is normalized into the same scale, which covers a 3-meter by 3-meter square in the actual scene according to the perspective map. Therefore, the first step of our nonparametric fine-tuning method focuses on retrieving training scenes that have similar perspective maps with the target scene from all the training scenes. Those retrieved scenes are called candidate fine-tuning scenes. A perspective descriptor is designed to represent the view angle of each scene. Since the perspective map is linearly fitted along the y axis, we use its vertical gradient ΔM_y as the perspective descriptor.

Based on the descriptor, for a target unseen scene, the top 20 perspective-map-similar scenes are retrieved from the whole training dataset as shown in Figure 4 (a). The retrieved images are treated as the candidate scenes for local patch retrieval.

4.2. Local patch retrieval

The second step is to select similar patches, which have similar density distributions with those in the test scene, from candidate scenes. Besides the view angle and the scale, the crowd density distribution also affects the appearance pattern of crowds. Higher density crowd has more severe occlusions, and only heads and shoulders can be observed. On the contrary, in sparse crowd, pedestrian appear with entire body shapes. Some instances of input patches are shown in Figure 4 (b). Therefore, we try to predict the density distribution of the target scene and retrieve similar patches that match the predicted target density distribution from the candidate scenes. For example, for a crowd scene with high densities, denser patches should be retrieved to fine-tune the pre-trained model to fit the target scene.

With the pre-trained CNN model presented in Section 3.2, we can roughly predict the density and the total count for every patch of the target image. It is assumed that patches with similar density map have similar output through the pre-trained model. Based on the prediction result, we compute a histogram of the density distribution for the target scene. Each bin is calculated as

$$c_i = \lfloor \ln(\hat{y}_i + 1) \times 2 \rfloor . \quad (4)$$

where $\hat{y}_i$ is the integrating count of estimated density map for sample i. Since there rarely exist scenes where more than 20 pedestrians stand in a 3-meter by 3-meter square, when $\hat{y}_i > 20$, the patch should be assigned to the sixth bin, i.e. $c_i = 6$. Density distribution of the target scene can be obtained from Equation (4). Then, patches are randomly selected from the retrieved training scenes and the number of patches with different densities are controlled to match the density distribution of the target scene. In this way, the proposed fine-tuning method is adopted to retrieve the patches with similar view angles, scales and density distributions. The fine-tuned crowd CNN model achieves better performance for the target scene. The results will be shown in the following section.

5. Experiment

We evaluate our method in three different datasets including our proposed the WorldExpo'10 crowd counting dataset, the UCSD pedestrian dataset [4] and the UCF_CC_50 dataset [8] . The details of the three datasets are described in Table 1 and example frames are shown in Figure 5.

5.1. WorldExpo'10 crowd counting dataset

We introduce a new large-scale cross-scene crowd counting dataset. To the best of our knowledge, this is the largest dataset focusing on cross-scene counting. It includes 1132 annotated video sequences captured by 108 surveillance cameras, all from Shanghai 2010 WorldExpo[2]. Since most of the cameras have disjoint bird views, they cover a large variety of scenes. We labeled a total of 199,923 pedestrians at the centers of their heads in 3,980 frames. These frames are uniformly sampled from all the video sequences. The details are listed in Table 1 and some instances are shown in Figure 5.

[2]Since most exhibition pavilions have been deconstructed, and no video corresponding to those pavilions still in use is included, the data is approved to be released for academic purposes.

Table 1. Statistics of three datasets: N_f is the number of frames; N_c is the number of scenes; R is the resolution; FPS is the number of frames per second; D indicates the minimum and maximum numbers of people in the ROI of a frame; T_p is total number of labeled pedestrians

Dataset	N_f	N_c	R	FPS	D	T_p
UCSD	2000	1	158*238	10	11-46	49885
UCF_CC_50	50	50	–	image	94-4543	63974
WorldExpo	4.44 million	108	576*720	50	1-253	199923

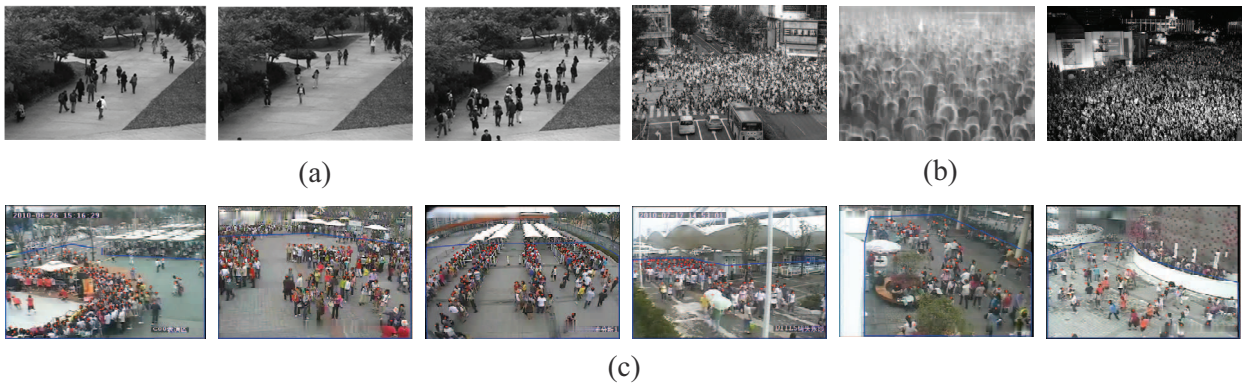


Figure 5. (a) Example frames of the UCSD dataset. (b) Example frames of the UCF_CC_50. (c) Example frames of the WorldExpo dataset. The region within the blue polygons are the regions of interest (ROI) and positions of pedestrian heads are labeled with red dots

Our dataset is splitted into two parts. 1,127 one-minute long video sequences out of 103 scenes are treated as training and validation sets. The test set has 5 one-hour long video sequences from 5 different scenes. There are 120 labeled frames in each test scene and the interval between two frames is 30 seconds. The pedestrian number in the test set changes significantly over time ranging from 1-220. The existence of large stationary groups makes it hard to detect the foreground area. Thus, most of the proposed counting methods are not applicable to our dataset, because their methods heavily rely on the segmentation of foreground.

The quantitative results of cross-scene crowd counting on our dataset are reported in Table 2. The Mean Absolute Error (MAE) is employed as the evaluation metric. Firstly, we attempt to extract LBP features and use the ridge regressor (RR) to estimate the crowd number, and the results are listed at the top row. The results predicted from our CNN crowd model without fine-tuning are shown at the second row. Then the results of our proposed method with data-driven fine-tuning are listed at the third row. These three methods do not use any data from the test scene. Our crowd CNN model can estimate density maps and crowd counts effectively. The data-driven fine-tuning method improves the performance in some test scenes. Similar samples retrieved from training data can help the model to better fit the test data. The density estimation results are shown in Figure 6.

We also observe that some auxiliary labeling in the target scene could boost the performance of our method. As scene-specific information is introduced, most background noise could be eliminated. Our predicted density map can be treated as feature and ridge regression is used to fit the pedestrian number. For comparison, we test two scene-specific methods in [6] and [7]. [6] is a global regression method using various hand-crafted features including area, perimeter, edge and local texture feature, while [7] adopts the random regression forest to predict the density map. The compared methods are trained with the first 60 labeled frames for every test scene, and the remaining frames are used as the test set. A GMM-based background modeling method is adopted to extract the foreground segments. Since a mount of stationary crowds exist in scene 2, it is hard to obtain foreground accurately. Our cross-scene crowd counting method outperforms the scene-specific methods. The results are further improved for test scene 1, scene 3 and scene 4 shown in Table 2. However, for scene 2, the ridge regression leads to a worse result, because the density distribution in the first 60 training frames have significantly differences with the remaining test frames.

We also compare our iterative switchable learning scheme with the joint multi-task scheme. The joint multi-task loss L_J is defined as:

$$L_J(\Theta) = L_D(\Theta) + \lambda L_Y(\Theta) \tag{5}$$

The average mean absolute errors of the two different losses in the proposed cross-scene WorldExpo'10 dataset are shown in Table 3. Our iterative switchable training process achieves better performance than the joint multi-task loss. Different but related objectives can help each other to

Table 2. Mean absolute errors of the WorldExpo'10 crowd counting dataset

Method	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Average
LBP+RR	13.6	58.9	37.1	21.8	23.4	31.0
Crowd CNN	10.0	15.4	15.3	25.6	4.1	14.1
Fine-tuned Crowd CNN	9.8	14.1	14.3	22.2	3.7	12.9
Luca Fiaschi et al. [7]	2.2	87.3	22.2	16.4	5.4	26.7
Ke et al. [6]	2.1	55.9	9.6	11.3	3.4	16.5
Crowd CNN+RR	2.0	29.5	9.7	9.3	3.1	10.7

Table 3. Average mean absolute errors (AMSE) on WorldExpo'10 crowd counting dataset via switching training scheme and the multi-task training scheme

t	1	2	3	4	5	6
AMSE	17.4	15.5	14.9	14.3	14.1	14.3
λ	10	1	0.1	0.05	0.01	0.005
AMSE	50.8	50.8	18.5	15.5	15.3	15.5

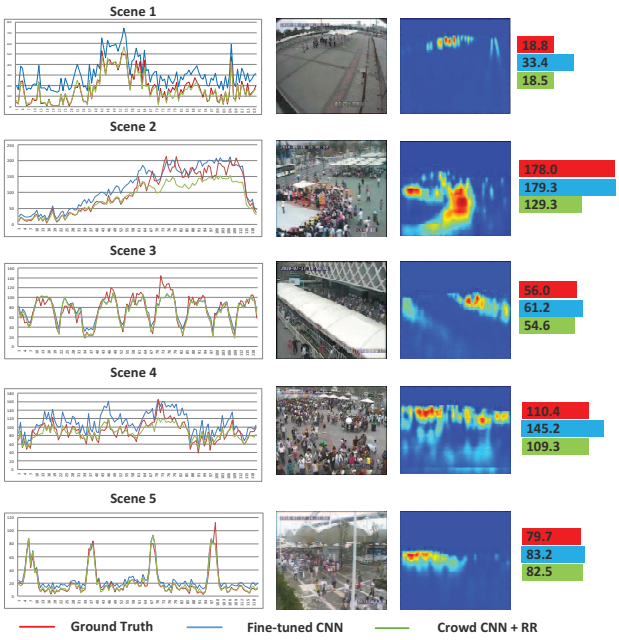


Figure 6. Our density estimation and counting results on the WorldExpo'10 crowd counting dataset. (Left) result curve for each test scene, where X-axis represents the frame index and Y-axis represents the counting number. (Middle) one sample selected from the corresponding test scene. (Right) density map and crowd estimated on the sample. **Best viewed in color**.

obtain better local optima through switching training objectives. In contrast, the joint multi-task scheme requires more computation to obtain a optimal λ than our switchable training process, and the results are also sensitive to the choice of λ.

5.2. UCSD dataset

Our second experiment focuses on crowd counting for a single scene. Our crowd CNN model is compared with scene-specific methods. A 2000-frame video dataset [4] is chosen from one surveillance camera in the UCSD campus. The video in this dataset was recorded at 10 fps with a frame size of 158×238. The labeled ground truth is at the center of every pedestrian. The ROI and perspective map are provided in the dataset.

We follow the dataset setting in [4] and employ frames 601-1400 as the training data and the remaining 1200 frames as test set. 72×72 patches are extracted from the image without normalization. 800 patches are randomly cropped from each image to train the model. For the test set, the patches are extracted in a sliding window fashion with 50% overlap. The density estimation of each pixel is obtained by averaging all the predicted overlapping patches. Our predicted density map from the CNN model can be treated as feature. The ridge regression is used to fit the training set.

Table 4. Comparison with global regression methods for crowd counting on the UCSD dataset

Method	MAE	MSE
Kernel Ridge Regression [1]	2.16	7.45
Ridge Regression [6]	2.25	7.82
Gaussian Process Regression [4]	2.24	7.97
Cumulative Attribute Regression [5]	2.07	6.86
Our Crowd CNN Model	**1.60**	**3.31**

Table 4 reports the errors with our methods and four other methods based on global regression. Two metrics, the MAE and Mean Squared Error (MSE), are employed for evaluating the performance of compared methods. Our proposed crowd CNN model outperforms all the global regression based approaches for both metrics. Note that our method does not rely on any foreground information and is tested on the whole area of ROI. Yet other compared methods rely on the foreground segmentation features. The methods we compared in Table 4 adopt similar hand-crafted features including segmentation features (area and perimeter), edge features obtained with the canny operator and local texture features (such as LBP [19] and GLCM [18]). The experiment results show that by incorporating the additional density information, our crowd CNN model boosts the accuracy of crowd counting significantly.

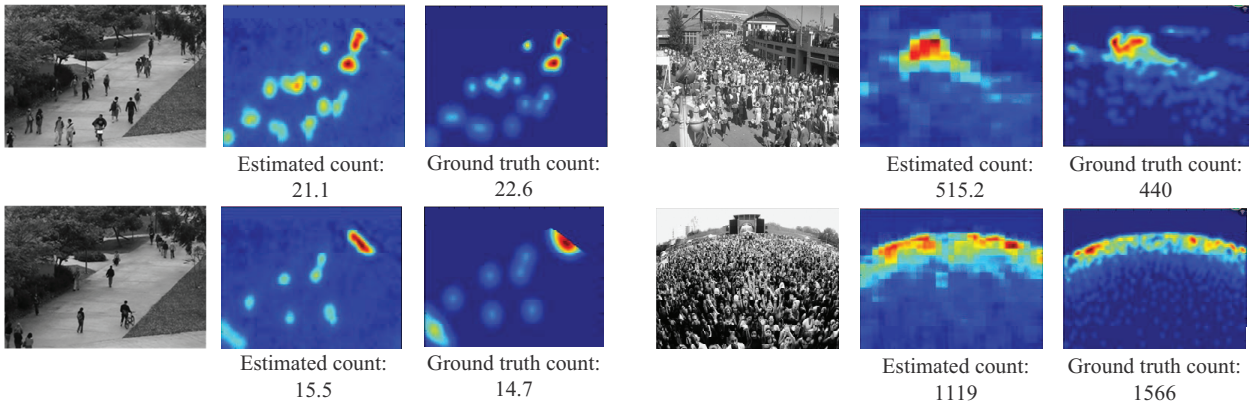


Figure 7. Density estimation results in the UCSD dataset and the UCF_CC_50 dataset. (Left) the input frame. (Middle) the predict result through our method. (Right) the density map ground truth

We compare our method with other density regression based methods in Table 5. Following the experiment settings in [12] and [24], we split the data into four different training and test sets: 1) 'maximal' :training on frames 600:5:1400; 2) 'downscale': training on frames 1205:5:1600; 3) 'upscale' : training on frames 805:5:1100; 4) 'minimal' : training on frames 640:80:1360. The frames outside the training range are tested. The four splits differ in the number of training images and the average number of pedestrians. Our method is comparable with state-of-the-arts. Again, unlike other density regression methods, our method does not require foreground segmentation. Some of our results are shown in Figure 7.

Table 5. Mean absolute errors of density regression methods and our approach on the UCSD dataset

Method	'max'	'down'	'up'	'min'
Density + RF [7]	1.7	2.16	1.61	2.2
Density + MESA [12]	1.7	1.28	**1.59**	2.02
Codebook + RR [2]	**1.24**	1.31	1.69	**1.49**
Our Crowd CNN Model	1.70	**1.26**	**1.59**	1.52

5.3. UCF_CC_50 dataset

The UCF_CC_50 dataset [8] contains images collected from Internet. It is a challenging dataset, because there are only 50 images in the dataset with pedestrian numbers ranging between 94 and 4543. The authors provided the labeled ground truth, which can be used to generate the ground truth density map as the right column of Figure 7.

Following by the dataset setting in [8], we split the dataset randomly and perform 5-fold cross-validation. MAE and MSE are employed as the evaluate metrics. Similar to the experimental setting in the USCD dataset, 1600 patches are randomly cropped from each image for training. The patch size is 72×72. The test patches are densely selected with 50% overlaps. The predicted density at each pixel is calculated by averaging overlapping prediction patches.

Table 6. Comparision results in UCF_CC_50 dataset

Method	MAE	MSE
Rodriguez et al [22]	655.7	697.8
Lempitsky et al. [12]	493.4	**487.1**
Idrees et al. [8]	468.0	590.3
Our Crowd CNN Model	**467.0**	498.5

We compared three methods on the UCF_CC_50 dataset. The methods presented in [12] proposed the MESA-distance to learn a density regression model using dense SIFT features [16] on randomly selected patches. Rodriguez *et al.* [22] made use of density map estimation to improve the head detection results in crowd scenes. Idrees *et al.* [8] relied on multi-source feature, including head detection, SIFT and Fourier analysis. There is no post-processing for all the compared methods. The experimental results are shown in Table 6. Our proposed method achieves the best MAE and is effective on cross-scene counting, even with the very tough test set. Some experimental results are shown in Figure 7. Still, our method can generate a reasonable density map and obtain a reasonable counting result close to the ground truth.

6. Conclusion

In this work, we propose to solve the cross-scene crowd counting problem with deep convolution neural network. The learned deep model specifically has better capability for describing crowd scenes than other hand-craft features. We propose a switchable training scheme with two related learning objectives, estimating density map and global count. With the proposed alternative training scheme, the two related tasks assist each other and achieve lower loss. Moreover, a data-driven method is proposed to select samples from the training data to fine-tune the pre-trained CNN model adapting to the unseen target scene.

7. Acknowledgement

This work is partially supported by NSFC (No. 61025005, 61129001, 61221001, 61301269), STCSM (No. 14XD1402100, 13511504501), 111 Program (No. B07022), Sichuan High Tech R&D Program (No. 2014GZX0009), General Research Fund sponsored by the Research Grants Council of Hong Kong (No. CUHK419412, CUHK417011, CUHK14206114, CUHK14207814), Hong Kong Innovation and Technology Support Programme (No. ITS/221/13FP) and Shenzhen Basic Research Program (No. JCYJ20130402113127496).

References

[1] S. An, W. Liu, and S. Venkatesh. Face recognition using kernel ridge regression. In *CVPR*, 2007.

[2] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman. Interactive object counting. In *ECCV*. 2014.

[3] G. J. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *CVPR*, 2006.

[4] A. B. Chan, Z. S. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *CVPR*, 2008.

[5] K. Chen, S. Gong, T. Xiang, Q. Mary, and C. C. Loy. Cumulative attribute space for age and crowd density estimation. In *CVPR*, 2013.

[6] K. Chen, C. C. Loy, S. Gong, and T. Xiang. Feature mining for localised crowd counting. In *BMVC*, 2012.

[7] L. Fiaschi, R. Nair, U. Koethe, and F. A. Hamprecht. Learning to count with regression forest and structured labels. In *ICPR*, 2012.

[8] H. Idrees, I. Saleemi, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. In *CVPR*, 2013.

[9] S. Jing, K. Kai, L. Chen, Chang, and W. Xiaogang. Deeply learned attributes for crowd scene understanding. In *CVPR*, 2015.

[10] K. Kai and W. Xiaogang. Fully convolutional neural networks for crowd segmentation. *arXiv preprint arXiv:1411.4464*, 2014.

[11] D. Kong, D. Gray, and H. Tao. A viewpoint invariant approach for crowd counting. In *ICPR*, 2006.

[12] V. Lempitsky and A. Zisserman. Learning to count objects in images. In *NIPS*, 2010.

[13] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014.

[14] Z. Lin and L. S. Davis. Shape-based human detection and segmentation via hierarchical part-template matching. *TPAMI*, 32(4):604–618, 2010.

[15] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *TPAMI*, 33:978–994, 2011.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[17] C. C. Loy, S. Gong, and T. Xiang. From semi-supervised to transfer counting of crowds. In *ICCV*, 2013.

[18] A. Marana, L. d. F. Costa, R. Lotufo, and S. Velastin. On the efficacy of texture analysis for crowd monitoring. In *SIBGRAPI*, pages 354–361. IEEE, 1998.

[19] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24(7):971–987, 2002.

[20] W. Ouyang and X. Wang. Joint deep learning for pedestrian detection. In *ICCV*, 2013.

[21] V. Rabaud and S. Belongie. Counting crowded moving objects. In *CVPR*, 2006.

[22] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert. Density-aware person detection and tracking in crowds. In *ICCV*, 2011.

[23] M. Rodriguez, J. Sivic, I. Laptev, and J. Audibert. Data-driven crowd analysis in videos. In *ICCV*, 2011.

[24] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Crowd counting using multiple local features. In *DICTA*, 2009.

[25] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[26] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*. Springer, 2010.

[27] M. Wang and X. Wang. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *CVPR*, 2011.

[28] N. Wang and D.-Y. Yeung. Learning a deep compact image representation for visual tracking. In *NIPS*, 2013.

[29] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV*, 2005.

[30] X. Zeng, W. Ouyang, M. Wang, and X. Wang. Deep learning of scene-specific classifier for pedestrian detection. In *ECCV*. 2014.

[31] X. Zeng, W. Ouyang, and X. Wang. Multi-stage contextual deep learning for pedestrian detection. In *ICCV*, 2013.