Crowd Density Estimation Based on a Modified Multicolumn Convolutional Neural Network

1st Wei-Teng Weng
Dept. of Computer Science and Information Engineering
National Taipei University

2nd Daw-Tung Lin*
Dept. of Computer Science and Information Engineering
National Taipei University

***Abstract*—Crowd management has been a topic of concern for many years because accidents frequently occur in situations with a high crowd density. With only a finite amount of space available during shows, protests, or other special occasions, a high crowd density can present a clear danger for those in the area. Considering these challenges, we employed and modified a three-tier multicolumn convolutional neural network (MCNN) system architecture to precisely estimate crowd density. We distinguished three regions from the near to far field to produce a crowd density map. Based on the MCNN system architecture, we detected changes in the size of a crowd according to a distance measure and examined additional features that can be incorporated to demonstrate their effects on crowd density maps. Examining these features using the Shanghaitech dataset demonstrated that compared with the native MCNN, the accuracy of estimating crowd counting by using our proposed method increased by 22.97% and 18.64% in terms of mean absolute error (MAE) and mean square error (MSE), respectively. A performance comparison with other state-of-the-art methods was also made. From this, we can infer that the proposed system is compatible with the other listed methods and is worthy of further investigation.**

***Index Terms*—Crowd Density Estimation, Crowd Counting, Convolutional Neural Networks, Muticolumn Convolutional Neural Network.**

I. Introduction

Crowd disasters usually occur in high-density crowds. During accidents, high-density crowds often result in more severe casualties. Therefore, crowd detection and crowd density estimation are useful for the surveillance of overcrowding situations. These parameters can enable security units to be notified regarding areas with potentially dangerous high-density crowds to prevent accidents or evacuate people when accidents occur. Many researchers have investigated the complications faced when attempting to count crowds based on image texture features, appearance, video tracing, or hardware detection methods [1]. Crowd counting approaches can be classified into three categories: 1) pedestrian detection, 2) pedestrian tracking and analysis, and 3) feature-based regression. Pedestrian detection is associated with people detection in crowds and is inherently challenging because a crowd or dense scene usually has numerous occlusions [2], [3]. Pedestrian tracking uses the correlation of continuous images in a video to estimate the crowd count. For example, the features from accelerated segment test, scale-invariant feature transform, robust local optical flow, and Bayesian clustering algorithms are some of the methods belonging to the pedestrian tracking category [4]–[6]; however, these methods rely on a sequence of images and cannot be applied to static images. Feature-based regression is one of the most popular methods for crowd counting. This method consists of three steps: 1) foreground segmentation, 2) foreground feature extraction, and (3) crowd count estimation using regression function. The extracted features are divided into categories such as texture, human size, edge, and shape. Because these features are independent, a classifier is required to identify the desired objective. Typical classifiers are a support vector machine (SVM) or Gaussian process regression. Fradi *et al.* [7] segmented images into several blocks and then examined these parts by using a local binary pattern (LBP) histogram. After extracting the histogram as a texture feature, the SVM is used for classification. Song *et al*. [8] applied binary edge features to calculate the number of people and then utilized an SVM classifier to perform intensive classification. Song *et al*. [9] combined different texture features, such as gray-level co-occurrence, LBP, and gradient orientation co-occurrence matrix, with different block sizes to estimate the number of people through Gaussian process regression. Lamba *et al*. [10] proposed a hybrid model containing Fourier analysis, LBP, gray-level dependence matrix, and a histogram of oriented gradient to detect heads and calculate the number of people. Mousse *et al*. [11] constructed a new surveillance system to calculate the number of people in a crowd scene by using overlapping cameras. The tracking and calculation of the number of people were performed by examining the association between each camera. Nevertheless, the aforementioned methods all rely on appropriate selection of features. In early days, feature selection was usually based on each researchers prior experience and experiments; however, no guarantee exists that these features are the most suitable. In this study, we adopted an end-to-end deep learning mechanism to train a network to automatically select the most appropriate features.

A deep neural network is a deep nonlinear network structure performing the approximation of complex functions. The gradient descent method is adopted to correct the weights to reduce the error of the neural network by using the back-propagation (BP) algorithm. In the calculation of the error gradient δ, the BP algorithm performs partial differentiation layer by layer. The deep learning of the convolutional neural network (CNN) architecture was published by Lecun *et al*. [12] in 1989; this is the first trained multilayer network structure successfully applied to digit recognition. In 2014, Chatfield *et*

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

al. [13] developed VGG Net, another CNN architecture that achieved a 13.1% error rate in the ILSVRC-2012 dataset (Top-5) competition. CNN has been proven to achieve favorable results not only in classifying image types but also in other applications [14]–[17]. Herein, we review the literature on crowd counting and crowd density analysis. Pu *et al*. [18] proposed a deep CNN (ConvNet) to divide people in an image into 3- or 5-class overall accuracy depending on crowd density. Wang *et al*. [19] applied CNN to analyze images and determine crucial features; thus, a crowd image can be enhanced and the number of people can be estimated. Liu *et al*. [20] proposed that crowd population can be calculated by the collecting statistics of CNN features; their network showed a favorable performance in counting people by adopting pre-trained parameters in conjunction with the SVM. Zhang *et al*. [21] segmented images into small pieces by using features extracted from a CNN and then combined CNN density maps to count the number of people for cross-scenes. Boominathan *et al*. [22] presented a two-layer CNN deep learning model with a shallow structure and then combined the density map output. The estimated density map was then used for crowd counting by calculating the integral areas of the density map. Zhang *et al*. [23] proposed that crowd count estimation should be generated using the correlation between density maps and the number of people. To avoid the effect of environmental changes on error rate, they considered the problems presented by an image in terms of their distance measure. Recently, Sindagi and Patel [24] proposed a novel end-to-end cascaded CNN networks to jointly learn crowd count classification and density map estimation by incorporating a high-level prior into the density estimation network and learning globally features to refine the density maps. Sam *et al*. [25] proposed a switching convolutional neural network using independent CNN regressors as local receptive fields and a switch classifier is trained to relay the crowd scene patch to the best CNN regressor. Zeng *et al*. [26] proposed a novel multi-scale convolutional neural network (MSCNN) based on the multi-scale blobs such that the network is able to generate scale-relevant features for higher crowd counting performances in a single-column architecture. Sindagi and Patel [27] further presented a Contextual Pyramid CNN (CP-CNN) by incorporating global and local contextual information of crowd images to generate high-quality crowd density and count estimation. The proposed method achieves lower count error and better quality density maps as compared to the recent state-of-the-art methods.

II. Methodology

In this paper, we propose a crowd counting network that can accurately estimate crowd density and draw estimated density maps for different scenes and perspectives. From the network, we can estimate the number of people and obtain a crowd distribution. By examining the color depth of a map, we can also identify whether a region is crowded. Violence and abnormal behaviors often occur in high-density regions; thus, this crowd counting network can be used to predict the probability of accidents in a region through analyzing the regions density maps

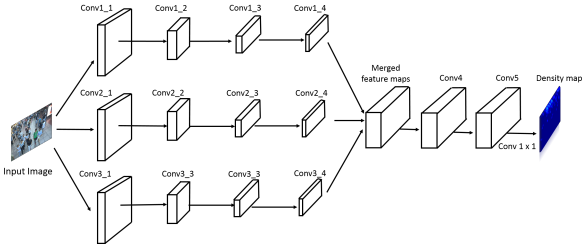


Fig. 1. Proposed crowd density estimation multicolumn convolutional neural network architecture.

A. Crowd Counting Network Architecture

The architecture of the proposed crowd counting network is based on the modified MCNN architecture [23] by adding additional two convolutional layers between the merge layer and the fully convolutional network (FCN) layer (Fig. 1). The two convolutional layers are added to strengthen the features of combined networks through the convolutional layers. Density map estimation is performed using the crowd counting network when an input image is provided, and the deep learning network updates weighing parameters by comparing the difference between the ground-truth density map and the currently estimated density map. Finally, the number of people in a scene is obtained from the integral of the estimated density map. In an MCNN, two layers of 2×2 max pooling exist for downsampling; however, in the proposed crowd counting network, we have added two convolutional layers, with each having a kernel size of 5×5. The first and second layers output 128 and 256 feature maps, respectively, which finally enter the FCN layer; all convolutional layers employ the rectified linear unit activations. Next, CNN loss functions, such as softmax, cross entropy, and Euclidean distance, can be chosen to generate the output. We found that the choice of loss functions affected the final results. Based on those results, the Euclidean distance is used as the loss function to calculate the difference between the estimated and ground-truth density maps. The loss function is defined as follows:

$$L(\alpha) = \frac{1}{2N} \sum_{i=1}^{N} \parallel F_i - D_i(x) \parallel_2^2, \tag{1}$$

where α denotes the learning rate, $D_i(x)$ is the i^{th} ground-truth density map, N is the total number of samples, and F_i is the i^{th} density map estimated using the network. The error calculated from $L(\alpha)$ is used to update network parameters during training. Then, the crowd count and distribution are obtained from the estimated density map.

B. Weights Initialization

The initial weights are crucial to a CNN because weight initialization can affect convergence results. A common technique for weight initialization is to use a Gaussian distribution;

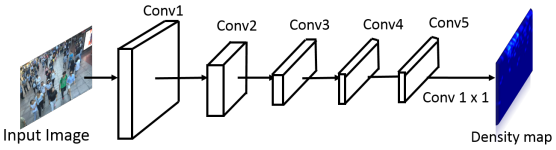


Fig. 2. One single branch of the proposed crowd estimation network structure.

however, we found that a large input image leads to a large loss when using a Gaussian distribution for weight initialization, which can make the network fall into a local minimum and cause the network training procedure to fail (even adjusting the learning rate was unable to solve this problem). Consequently, we adopted the normalized initialization method for weight initialization proposed by Glorot *et al*. [28] to achieve smooth convergence. The weighting parameter W between the layers j and $j+1$ was based on the normalized initialization method as follows:

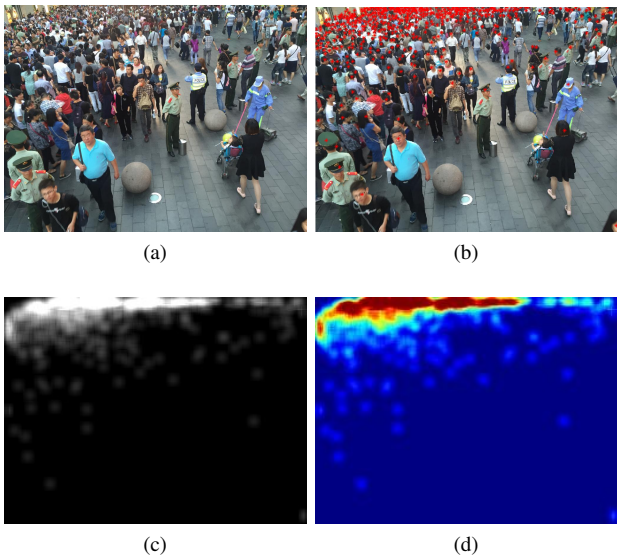
$$W \sim U[-\sqrt{\frac{6}{n_j + n_{j+1}}}, \sqrt{\frac{6}{n_j + n_{j+1}}}], \qquad (2)$$

where n_j and n_{j+1} denotes the size of the layers j and $j+1$, respectively.

To leverage the proposed MCNN to obtain a favorable training convergence, we pretrained the three branches of MCNN (Fig. 1) separately by adding one FCN layer and 1×1 convolutional layer, as illustrated in Fig. 2. We were then able to fine-tune the trained branches and put them back into the MCNN architecture. By using this method, we could improve the convergence and accuracy of the network.

C. Ground-Truth Density Maps Generation

The CNN architecture is a supervised learning approach; therefore, the ground truths of the corresponding training data are required. The purpose of the proposed network is to train and estimate the corresponding density maps. The quality of the ground-truth density map affects the convergence of the network architecture. In a given crowd image, the proportion of long- and close-shot scenes can be altered due to perspective distortion. Perspective distortion is determined by relative distances at which an image is taken. Because the head distance of pedestrians is relevant to the distance, the head distance of a crowd must be set for each image. First, we manually located and marked the head positions $P(x)$ of pedestrians. The head position was determined to be at the center of the head and plotted in red dots as demonstrated in Fig. 3(b). Next, the density map $D(x)$ was obtained by computing the convolution of $P(x)$ and Gaussian kernal $G_\sigma(x)$: $D(x) = P(x) * G_\sigma(x)$ [29], [30], where $\sigma = \beta d$ and d denotes the head distance. We first fixed the value of d, and the parameter β was set to 0.3 in accordance with Zhang *et al*. [23] to obtain high-accuracy density maps. Then, the ground-truth density map was established as a gray-level image exhibited in Fig. 3(c).



(a) (b) (c) (d)

Fig. 3. (a) Original input image; (b) head positions are marked in red dots; (c) grey-level density map; and (d) color density map.

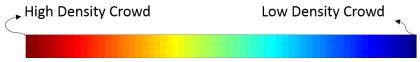
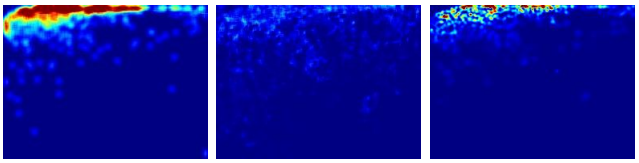


Fig. 4. Density color distribution

For ease of visualization, we applied a normalization method to colorize the density map (Fig. 4). Different colors were used to represent crowd density, where red and blue represented the highest and lowest densities, respectively. The resultant color version density is illustrated in Fig. 3(d). During the training process, Fig. 3(a) was used as the input image and Fig. 3(d) as the ground-truth density map.

The result of density map estimation is presented in Fig. 5(b). However, we observed that the initial estimated density map Fig. 5(b) did not match with the ground-truth density map Fig. 5(a). Through examination, we concluded that crowd distribution was relevant to a high-density area. Thus, we modified the head distance as the average distance of the five nearest head positions: $\bar{d} = \frac{1}{5}\sum_{n=1}^{5} d_n$. We called this approach as adaptive head distance ($\bar{d}$). As shown in Fig. 5(c), the use of the average of the five nearest head distances resulted in a more accurate density map, and the distribution



(a) (b) (c)

Fig. 5. (a) Ground-truth density map; (b) estimated density map with fixed d; and (c) estimated density map with adaptive $\bar{d}$.

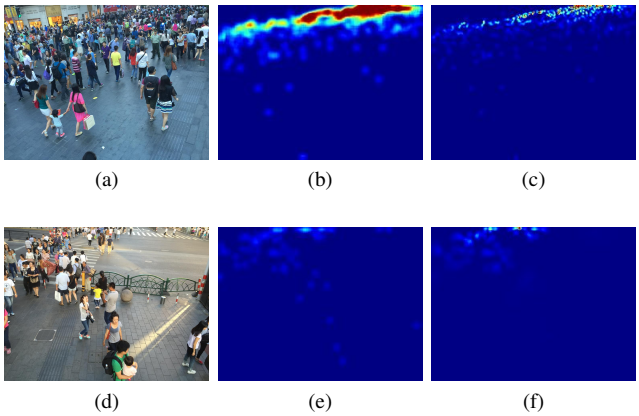


Fig. 6. (a) Original density map; (b) density map using fixed $\overline{d}$; (c) density map using adaptive $\overline{d}$; (d) original density map; (e) density map using fixed $\overline{d}$; and (f) density map using adaptive $\overline{d}$.

of crowds in a high-density area could be detected using adaptive $\overline{d}$. More experimental results are presented in Fig. 6, demonstrating that the estimated density maps using adaptive $\overline{d}$ (Fig. 6(c) and (f)) are better than the density maps using fixed d (Fig. 6(b) and (e)).

D. Fitting the Input Image: Padding and Cutting

In this study, the CNN input layer size was set to 1024×768. However, the dataset being used (Shanghaitech dataset A) had different input image sizes. Instead of resizing the dataset images, which may have caused distortion, we applied padding and cutting methods to fit the input images to the network input layer. Examples of modified testing images are shown in Fig. 7. If the training/testing image was smaller than the input layer specification, we filled and padded with zeros in the extra area. For example, as illustrated in Fig. 7(a), the image size 500×334 was smaller than 1024×768; thus, we padded zeros to the boundary of the original input image to extend the size to 1024×768. By contrast, if the image was larger than the input layer specification, we used the cutting method. The oversized image was cut based on the specified height and width of CNN input layer as a window starting from the top-left corner of the original image and slid the window to the right and bottom with the least overlapping between each cut as shown in Fig. 7(b). Chopped images were then input to the network for training or testing separately, and the resultant density maps were then combined accordingly.

III. Experimental Results

The proposed crowd counting system was implemented using the CAFFE deep learning platform on a computer equipped with an Intel Xeon-E5-2640 CPU@2.4GHz, Nvidia GTX 1080Ti GPU, Linux Ubuntu16.04 64bit OS and 16GB RAM. We applied the Shanghaitech dataset as training and testing data. The Shanghaitech dataset was first introduced by Zhang *et al*. [23] and has different scenes and views of the images. It contains 1198 images in crowded scenes and sparse scenes that include a total of 330,165 markers of heads

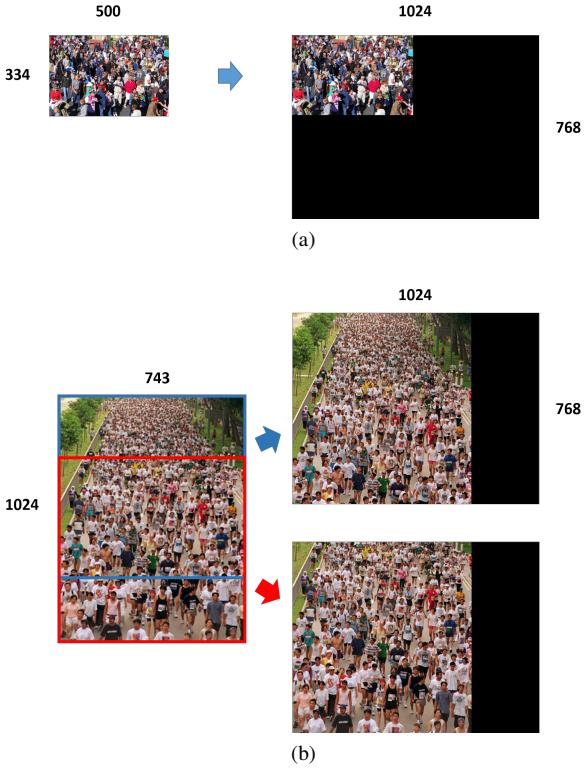


Fig. 7. Modified training/test images from the Shanghaitech dataset part A: (a) padding method and (b) cutting method.

and two parts, A and B. The images in part A were density scenes obtained from the Internet with different size images. These images included approximately 33∼3139 people. We adopted 300 images as training data and 182 images for testing from part A of the dataset. The images in part B were all 1024×768 color pictures that were taken on the street and included approximately 9∼578 people. We selected 400 images as training data and 316 images as testing data from part B. Detailed descriptions of the Shanghaitech dataset are listed in Table I, and examples of images are presented in Fig. 8.

To evaluate the performance of crowd density estimation, we applied mean absolute error (MAE) and mean squared error (MSE) computed as follows [21], [23]:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - x_i|, \tag{3}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - x_i)^2}, \tag{4}$$

where N is the number of total samples, y_i is the ground-truth value of the number of people in the image, and x_i is the estimated number of people by the CNN architecture. MAE represents the accuracy of the estimated number of people, and MSE represents the accuracy robustness of the estimated number of people.

TABLE I
DETAILED DESCRIPTIONS OF THE SHANGHAITECH DATASET.

Dataset	Image Size	Image Type	Training Images	Test Images	Max. Count	Min. Count	Avg. Count	Total Count
Part A	Different	Color, Gray	300	182	3139	33	501.4	241,677
Part B	1024×768	Color	400	316	578	9	123.6	88,488

Fig. 8. Examples of the Shanghaitech dataset images.

TABLE II
PERFORMANCE COMPARISON WITH AND WITHOUT ADAPTING HEAD DISTANCE d USING THE SHANGHAITECH DATAST PART B.

Shanghaitech Dataset Part B		
Method	MAE	MSE
Fixed d	41.02	64.44
Adaptive $\bar{d}$	**31.9**	**52.9**

First, we evaluated the network performance with head distance and adaptive head distance using the Shanghaitech dataset part B for same input image size. The results are listed in Table II. The system performed with substantially increased accuracy (an approximately 18% increase) when adaptive head distance was applied.

A performance comparison with other benchmark methods was also made (Table III). From this, we can infer that the proposed system outperforms the other listed methods. In contrast to MCNN, our method increased accuracy by 22.97% and 18.64% in terms of MAE and MSE, respectively. From the comparison of the estimated density maps with the ground-truth density maps (Fig. 9), we can observe that the estimated maps were clearly similar to the ground-truth maps. Table IV presents a comparison of crowd counting accuracy by using dataset part A with other methods. The proposed method had improved performance in comparison with those of Zhang *et al*. [21] and Zhang *et al*. [23]. The ground-truth density maps displayed a high degree of similarity with the estimated density maps (Fig. 10).

Recently, the deep learning CNN approaches for tackling crowd counting problems have achieved considerable progress.

TABLE III
PERFORMANCE COMPARISON WITH THE BENCHMARK METHODS USING THE SHANGHAITECH DATASET PART B.

Shanghaitech Dataset Part B		
Method	MAE	MSE
Zhang et al. [21]	30.3	48.7
MCNN Zhang et al. [23]	26.4	41.3
Ours	**20.6**	**33.6**

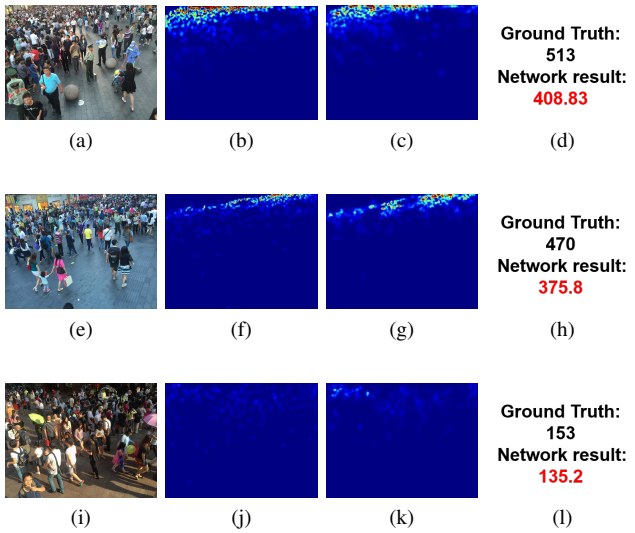


Fig. 9. Experimental results using the Shanghaitech dataset part B: (a) input image; (b) ground-truth density map; (c) result and estimation of density map; (d) crowd counting results; (e) input image; (f) ground-truth density map; (g) result and estimation of density map; (h) crowd counting results; (i) input image; (j) ground-truth density map; (k) result and estimation of density map; and (l) crowd counting results.

During the revision of this final version of paper, we learned that more advanced research results have just been published. Table V presents the performance comparison with the state-of-the-art methods using the Shanghaitech dataset. The proposed method outperforms the methods of Zhang *et al*. [21] and Zhang *et al*. [23] for both MAE and MSE measures. For dataset part B, the proposed method's MAE and MSE performance is compatible with those of Cascaded-MTL [24], Switch-CNN [25], MSCNN [26], and CP-CNN [27]. However, for dataset part A, our method still has room to be improved. Moreover, the most recent works published in the arXiv (a repository of electronic preprints) including CNN-MRF proposed by Han *et al*. [31], DAN presented by Li *et al*. [32], and Crowd Ranking Network proposed by Liu *et al*. [33] have accomplished further lower count error and better crowd

TABLE IV
PERFORMANCE COMPARISON WITH THE BENCHMARK METHODS USING THE SHANGHAITECH DATASET PART A.

Shanghaitech Dataset Part A		
Method	MAE	MSE
Zhang et al. [21]	181.8	277.7
MCNN Zhang et al. [23]	110.2	173.2
Ours	**108.2**	**171.3**

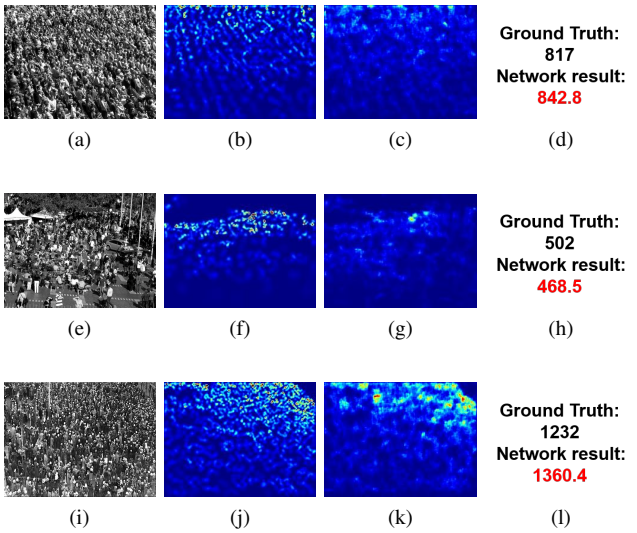


Fig. 10. Experimental results using the Shanghaitech dataset part A: (a) input image; (b) ground-truth density map; (c) result and estimation of density map; (d) crowd counting results; (e) input image; (f) ground-truth density map; (g) result and estimation of density map; (h) crowd counting results; (i) input image; (j) ground-truth density map; (k) result and estimation of density map; and (l) crowd counting results.

density estimation results.

IV. CONCLUSION

By using a three-tier CNN, we altered the method for establishing a ground-truth density map to increase self-adaptability to distinguish three regions from near to far. The changes to the weight initialization method improved the convergence of the network as well as increased the number of MCNN layers to extract high-level features and improve crowd estimate density map results. In this study, the Shanghaitech dataset part A and B was used for validation and analysis. From the experimental results of using Shanghaitech dataset part B, we can observe that with the use of original MCNN, the MAE and MSE error rates were 26.6 and 41.3, respectively; however, the use of our proposed method reduced the MAE and MSE error rates to 20.6 and 33.6, respectively. In other words, the proposed method increased the crowd counting estimation accuracy by 22.97% and 18.64% in terms of MAE and MSE, respectively. For Shanghaitech dataset part A, our method also outperforms the other state-of-the-art methods.

In future research, we hope to modify the framework to make its input image size unrestricted because the current limitations of the framework prevented us from using input images of all sizes. We will experiment with image sizes and perform different network adjustments to develop a network that can fit all image sizes and types. Finally, we will re-examine the number of branches in the MCNN architecture because the three branches that are currently used may not be optimal. Therefore, we will attempt various numbers of branches to achieve an optimized framework. For comprehensive assessment and validation, we will further conduct more experiments on the other three datasets: WorldExpo10 dataset [21], UCF_CC_50 dataset [34], and UCSD crowd-counting dataset [35].

TABLE V
PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART METHODS USING THE SHANGHAITECH DATASET.

Shanghaitech Dataset	Part A		Part B	
Method	MAE	MSE	MAE	MSE
Zhang et al. [21]	181.8	277.7	32.0	49.8
MCNN Zhang et al. [23]	110.2	173.2	26.4	41.8
Cascaded-MTL [24]	101.3	152.4	20.0	31.1
Switch-CNN [25]	90.4	135.0	21.6	33.4
MSCNN [26]	83.8	127.4	17.7	30.2
CP-CNN [27]	73.6	106.4	20.1	30.1
Ours	**108.2**	**171.3**	**20.6**	**33.6**
CNN-MRF [31]	79.1	130.1	17.8	26.0
DAN [32]	81.8	134.7	13.2	20.1
Multi-task (Query-by-example) [33]	72.0	106.6	14.4	23.8
Multi-task (Keyword) [33]	73.6	112.0	13.7	21.4

ACKNOWLEDGMENT

The authors would like to thank the Ministry of Science and Technology in Taiwan for partial support of research grant MOST 105-2221-E-305-006-MY3.

REFERENCES

[1] Y. Yuan, "Crowd monitoring using mobile phones," in *Sixth Int. Conf. on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 1, 2014, pp. 261–264.

[2] Z. Lin and L. S. Davis, "Shape-based human detection and segmentation via hierarchical part-template matching," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 604–618, 2010.

[3] M. Wang and X. Wang, "Automatic adaptation of a generic pedestrian detector to a specific traffic scene," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3401–3408.

[4] H. Fradi and J.-L. Dugelay, "Crowd density map estimation based on feature tracks," in *IEEE 15th Int. Workshop on Multimedia Signal Processing (MMSP)*, 2013, pp. 040–045.

[5] V. Rabaud and S. Belongie, "Counting crowded moving objects," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2006, pp. 705–711.

[6] G. J. Brostow and R. Cipolla, "Unsupervised bayesian detection of independent motion in crowds," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2006, pp. 594–601.

[7] H. Fradi, X. Zhao, and J.-L. Dugelay, "Crowd density analysis using subspace learning on local binary pattern," in *IEEE Int. Conf. on Multimedia and Expo Workshops (ICMEW)*, 2013, pp. 1–6.

[8] S. Hongquan, L. Xuejun, L. Guonian, Z. Xingguo, and W. Feng, "Video scene invariant crowd density estimation using geographic information systems," *China Communications*, vol. 11, no. 11, pp. 80–89, 2014.

[9] M. Saqib, S. D. Khan, and M. Blumenstein, "Texture-based feature mining for crowd density estimation: A study," in *Int. Conf. on Image and Vision Computing New Zealand (IVCNZ)*, 2016, pp. 1–6.

[10] S. Lamba and N. Nain, "Multi-source approach for crowd density estimation in still images," in *2017 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, Feb 2017, pp. 1–6.

[11] C. M. M. A. Mousse and E. C. Ezin, "People counting via multiple views using a fast information fusion approach," *Multimedia Tools and Applications*, pp. 6801–6819, Mar 2017.

[12] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[13] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[14] W. Ouyang, X. Zeng, X. Wang, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, H. Li, K. Wang, J. Yan, C. C. Loy, and X. Tang, "DeepID-Net: Object detection with deformable part based convolutional neural networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1320–1334, July 2017.

[15] S. Bianco, C. Cusano, and R. Schettini, "Single and multiple illuminant estimation using convolutional neural networks," *IEEE Trans. on Image Processing*, vol. PP, no. 99, pp. 1–1, June 2017.

[16] D. Ren, Y. Zhao, H. Chen, Q. Dong, J. Lv, and T. Liu, "3-D functional brain network classification using convolutional neural networks," in *2017 IEEE 14th Int. Symposium on Biomedical Imaging (ISBI 2017)*, April 2017, pp. 1217–1221.

[17] T. q. Peng and F. Li, "Image retrieval based on deep convolutional neural networks and binary hashing learning," in *2017 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 1742–1746.

[18] S. Pu, T. Song, Y. Zhang, and D. Xie, "Estimation of crowd density in surveillance scenes based on deep convolutional neural network," *Procedia Computer Science*, vol. 111, pp. 154–159, 2017.

[19] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proceedings of the 23rd ACM Int. conf. on Multimedia*, 2015, pp. 1299–1302.

[20] C. L. S. Liu, S. Zhai and J. Tang, "An effective approach to crowd counting with cNN-based statistical features," in *2017 Int. Smart Cities Conference (ISC2)*, Sept 2017, pp. 1–5.

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

[22] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "CrowdNet: A deep convolutional network for dense crowd counting," in *Proceedings of the 2016 ACM on Multimedia Conf.*, 2016, pp. 640–644.

[23] 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 Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 589–597.

[24] V. A. Sindagi and V. M. Patel, "CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting," in *2017 14th IEEE International Conf. on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017, pp. 1–6.

[25] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, no. 3, 2017, p. 6.

[26] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, "Multi-scale convolutional neural networks for crowd counting," in *Proceedings of the IEEE Conf. on Image Processing (ICIP)*, 2017, pp. 465–469.

[27] V. A. Sindagi and V. M. Patel, "Generating high-quality crowd density maps using contextual pyramid cnns," in *2017 IEEE International Conf. on Computer Vision (ICCV)*. IEEE, 2017, pp. 1879–1888.

[28] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." in *Aistats*, vol. 9, 2010, pp. 249–256.

[29] S. Cui, O. Meynberg, and P. Reinartz, "Bayesian linear regression for crowd density estimation in aerial images," in *Joint Urban Remote Sensing Event (JURSE)*, 2017, pp. 1–4.

[30] M. Á. Manso-Callejo, F.-K. K. Chan, T. Iturrioz-Aguirre, and M. T. Manrique-Sancho, "Using bivariate gaussian distribution confidence ellipses of lightning flashes for efficiently computing reliable large area density maps," *IEEE Trans. on Geoscience and Remote Sensing*, 2017.

[31] K. Han, W. Wan, H. Yao, and L. Hou, "Image crowd counting using convolutional neural network and markov random field," *arXiv preprint arXiv:1706.03686*, 2017.

[32] H. Li, X. He, H. Wu, S. A. Kasmani, R. Wang, X. Luo, and L. Lin, "Structured inhomogeneous density map learning for crowd counting," *arXiv preprint arXiv:1801.06642*, 2018.

[33] X. Liu, J. van de Weijer, and A. D. Bagdanov, "Leveraging unlabeled data for crowd counting by learning to rank," *arXiv preprint arXiv:1803.03095*, 2018.

[34] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2013, pp. 2547–2554.

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