Binary Mask based Crowd Counting analysis using Multi-Column Convolutional Neural Network

Lara Kašca and Ana Gavrovska, *Member, IEEE*

***Abstract*—Crowd counting has attracted significant attention recent years since it is valuable to estimate the number of people in a crowd in numerous applications, especially the ones related to video surveillance. Artificial intelligence, especially convolutional neural networks, became a part of such applications. In this paper, multi-column convolution neural network implementation has been analyzed, where the output is density map. The number of people is estimated as a sum of the map. In this paper experimental analysis using binary mask based postprocessing and ShanghaiTech dataset is performed. The obtained results seem promising in dealing with unwanted texture details related to irrelevant regions as in the case of greenery.**

***Index Terms*— Image processing, frame, crowd counting, density map, computer vision, MCNN, binary mask.**

I. Introduction

CROWD counting using a single image or a frame has been popular over the years [1-4]. It implies rather approximate techniques, where instead of determining the exact number of people in the crowds, estimation techniques can be applied. The need to develop these techniques arose due to the desire to find out how many people are at public gatherings such as rallies, traffic jams, disasters, protests or those mass events where this information could not be obtained based on the number of tickets sold. It can be considered useful for behavioral analysis when counting is performed in specific moments.

Counting all heads in a crowd image may be difficult or impossible. This is the reason why modern crowd counting concept relies on using a grid in order to segment the whole picture. The first step after counting the people manually in a few segments is to calculate the average number of people per segment after initial analysis. Then, the average number of people is multiplied by the total number of segments. This technique is called Jacobs' technique after journalism professor Herbert A. Jacobs at the University of California, Berkley. In the 1960s he applied this for counting students who protest the Vietnam War [1]. Also, he described a light and a dense crowd by cases having one person per 10 and 4.5 square feet area, meaning approximately 0.93 and 0.42 square meters area, respectively [1]. A slightly more sophisticated approach is to do an extrapolation adapted to the current part of the image, because not all parts are the same. However, the solutions are not linear and the mentioned approaches will not always give a satisfactory result. In 1995, one of the largest protests in American history, called the Million Man March, was held in Washington to raise public awareness of the position of African Americans in America. To this day, scientists are dealing with this event and how many people were in Presidential Park at the time, and estimates range between 400,000 and 1.1 million, which is a very large range giving inaccurate information [2].

Advances in technology today speak of much more elegant and impressively precise methods and calculations involving computer vision. Recent advances favorize convolutional neural network (CNN) in crowd counting [5-6]. In this paper we analyze some of the disadvantages of using a multi-column CNN. The experimental results presented in this paper shows the possibility to employ binary masks for CNN based refinement of the people counter which is in accordance to recent experiments with contextual representation [7-9].

The paper is organized as follows. After the introduction, a brief review regarding crowd counting methods is given. In Section II general usage of CNN is explained, as well as its application in crowd counting. Details regarding the simulation using binary mask approach and multi-column CNN performed in this paper are explained in Section III. The experimental results followed by discussion are presented in Section IV. Finally, in Section V conclusions are given.

II. Crowd counting and CNN

A. Crowd counting

Current methods from the literature related to crowd counting can be classified into one of the following four groups: detection-based methods, regression-based methods, density estimation-based methods and CNN-based methods [3-11]. In previous research on the CNN based topic, there are two main approaches. The first one is to pass an input image through the network architecture, and to give an estimation of the number of people as an output result. The second approach, used here, is to forward an image to the input of a network, and to get output density maps that needs to be processed further in order to get the final result. The reason

Lara Kašca is with the School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11020 Belgrade, Serbia (e-mail: larakasca@yahoo.com).

Ana Gavrovska is with the School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11020 Belgrade, Serbia (e-mails: anaga777@gmail.com; anaga777@etf.rs).

why this approach may seem better is that it provides more information than just one numerical value. The density map provides data on the density of different parts of analyzed image. Ground truth density map (GTDM) can be calculated using k nearest neighbor (KNN) method, and the corresponding neural network output is estimated density map (ESDM). An example of GTDM is presented in Fig. 1.

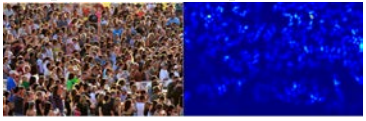

Fig. 1. Input image (left) and corresponding ground truth density map (right).

Density map used as a reference is determined using delta function δ at locations of interest convoluted with Gaussian kernel G in order to obtain a continuous representation for each image pixel x:

$$GTDM(x) = \sum_{i=1}^{N} \delta(x - x_i) * G\sigma_i(x). \qquad (1)$$

where x_i represents image pixel of interest where a person's head is found, N is the total number of people, and σ_i is variance proportional to mean distance value $d_{i,mean}$ for each x_i ($0.3*d_{i,mean}$). The mean distance, $d_{i,mean}$, is the average value of distances between each pixel x_i and its k nearest neighbors.

B. Convolutional Neural Network architecture

Convolutional neural network (CNN) is a neural network mostly used for visual data, such as images. As the name suggests, it uses the convolution operation in layers, usually in the first layers of a deep architecture. Input image is filtered (convoluted) by multiple filters, where CNN has the ability to represent images in more appropriate form having in mind relevant features. What can be applied in addition to the usual convolution is padding the input due to filtering application, or one can use strided convolutions that have the opposite effect to reduce the output. Convolutional layers are applied to learn features like edges using smaller segments of input data.

In addition to convolutional layers, there is also a pooling layer that is applied after the convolutional layer. It has the task of choosing max, sum or average value, and only passes that one value in order to further reduce the relevant data. The third type of layers that are optional are fully connected layers that usually take the last matrix output from convolutional-pooling layer sequence, and flattens it into a row or a vector. It can further pass it to the classic neural network. Fully connected layer often enables learning correlations between previous image representations called feature maps. Back-propagation is used for training via a number of iterations or epochs using available input and corresponding reference or target. The advantage of CNN application is a smaller number of parameters for determining and sharing, i.e. less feature engineering than in a standard NN concept.

In order to overcome the issue of filter size, instead of single-column, one may apply parallel CNN architecture with final layer for merging the results of each CNN branch. This can be particularly valuable for different person's head size in a crowd counting challenge. In Fig. 2, multi-column CNN (MCNN) architecture is illustrated. Moreover, binary image or binary mask (BM) may be used in pre- and/or post-processing to improve the estimation of the number of people [9]. Here, in Fig. 2, a post-processing block is presented, where final result is a density map where the mask removes irrelevant details from the estimated map.

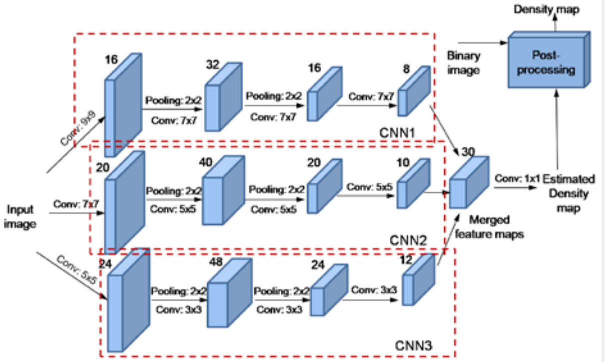

Fig. 2. Three CNNs making MCNN architecture with post-processing.

III. Simulation

This paper experiments with ShanghaiTech dataset of crowded images consisted of two subsets A and B [6]. The subset A contains 300 train and 182 test images, while the subset B has 400 train and 316 test images. In Fig. 3 the train examples are presented. The subset A is made up of images downloaded from the internet of densely distributed people, while they are less dense crowds found in subset B. The dataset statistics is presented in Table I. In addition to ShanghaiTech set, we also performed manual labeling of people in subset C for the purpose of testing, where crowd images were downloaded from internet.

Fig. 3. Examples of images from the train sets of subset A and B.

TABLE I
The Dataset Statistics

Subset (resolution)	Number of images	Min - max number of people (average)	Total number of people
A (Various)	482	33-3139 (501.4)	241677
B (768x1024)	716	9-578 (123.6)	88488
C (Various)	4	60-817 (322)	1288

The architecture in this paper is a multi-column CNN consisted of 3 parallel CNNs. It uses max pooling layers which apply to 2x2 regions and ReLU as an activation function. Images used in training are not used while testing. The network output is estimated density map (ESDM), where GTDM from (1) is used as input. The training is performed according to Euclidean distance between the maps.

A. Experimental crowd analysis

In the first phase of experimental analysis, MCNN is trained. Two subsets, A and B, are trained for single-image estimation. Metrics, like mean absolute error (MAE) and the mean square error (MSE), can be used for the training:

$$MAE = \frac{1}{M}\sum_{m=1}^{M}\left|p_m - p_{m,est}\right|,\ MSE = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left|p_m - p_{m,est}\right|} \quad (2)$$

where M is the number of images, p_m is the number of people in m-th image and $p_{m,est}$ is the estimated number of people in the same image. Here, the training is performed using MAE. CNN implementation and crowd analysis are performed using Python 3.7 and Pytorch [12], where for visualization purposes visdom is applied [13]. Used configuration is Intel i5 8400, Nvidia GeForce GTX 1050 Ti, cuda 10.1. For MCNN evaluation relative ratio is calculated:

$$R[\%] = \left|p_m - p_{m,est}\right| / p_m, \quad (3)$$

using A, B and C images. The number of people, p_m, can be considered in two ways, as a number of manual labels or as a sum of GTDM. The first way is used in the first phase, and the second way is used in the following phase.

In the second phase binary masks (BMs) for GTDM and ESDM are obtained by thresholding T (T=0 for GTDM and T=0.01 for ESDM). The thresholded ESDM can be considered as MCNN mask. In this paper, the proposed method for BM calculation for the post-processing consists of five steps: original image resizing, median residual calculation, extreme removal, local representation (histogram, histogram of gradients (HOG), sum) comparison, and morphological processing. In the first step, input image is resized according to ESDM, or MCNN output. The difference between the input and its median filtered version is calculated to obtain the median residual image. The next step is extreme removal, where the extreme details are considered to be the ones with the most highest values of difference and relative difference between the input and the filtered input. The fourth step consists of calculation of histogram, HOG and array of sums per rows and columns for the pixel neighborhood of size 5x5 (block size). Euclidean distances between the normalized histograms, the HOG and the sum representations of the blocks, are calculated and thresholded with 30%, 60% and 50-75% of the maximum values, respectively. Then, the intersection I of the three images is found, where only pixels with dense concentration within blocks are kept and used as seeds in the intersection image for region growing segmentation [14], obtaining binary image I_g. The final step is dilatation of I_g, where structuring element is square of size 15, obtaining BM1. With additional dilatation of size 10, the mask denoted as BM2 is obtained. In order to get the estimated number (est. num.) of people, the sum of density map is calculated according to the positions described by the binary mask, like BM1 or BM2. Also, the relative ratio according to (3) is calculated.

IV. Experimental results

In Fig. 4 MAE values through epochs are presented for training and testing in the case of A and B subset. Best performance is obtained for epochs 1715 and 939, respectively, where for further experiments the model based on A subset is used. The layout of the visdom window during training is shown in Fig. 5, where GTDM and ESDM for an image are presented. In Table II a comparison between obtained MAE based results and the results obtained in [6] is given showing slight variation. Besides ShanghaiTech dataset additional inputs are made according to manual labeling as shown in Fig. 6. In Table III calculated relative ratio results are presented for A, B and C image samples.

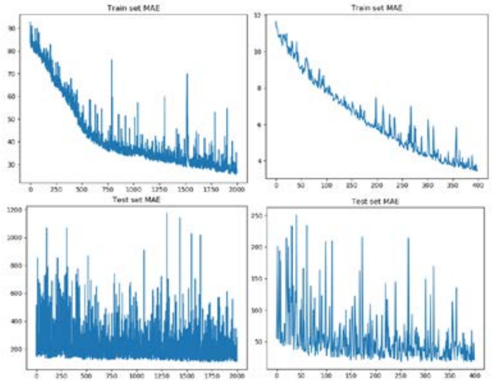

Fig. 4. *MAE values through epochs (on the left for subset A, on the right for subset B).*

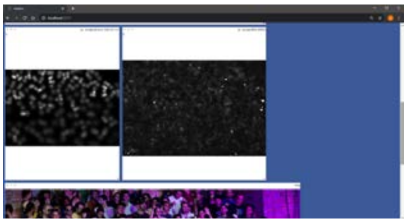

Fig. 5. Visdom window with GTDM and ESDM visualization.

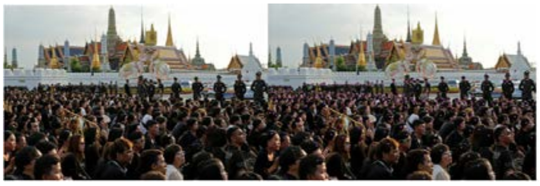

Fig. 6. Original (left) and manual labeling (right) of image1, Ctest taken from [15].

TABLE II
MAE for multi-column CNN

Subset/ Experiment	MAE (MSE) [6]	Obtained MAE in this paper
A	110.2 (173.2)	108
B	26.4 (41.3)	18.5

TABLE III
Relative ratio results for MCNN

Image No. (class)	Number of labels	MCNN count	Ratio R [%]
Image0,Atrain	1546	1420	8.2
Image0,Atest	172	598	**247.7**
Image3,Atest	211	654	**210.0**
Image5,Atest	431	435	0.9
Image5,Btest	57	92	61.4
Image24,Btest	70	64	8.6
Image1,Ctest	304	273	10.2
Image3,Ctest	817	557	31.8

It can be observed in Table III that MCNN count results are satisfying. Example of the successful density representations and superimposed maps and labels is shown in Fig. 7. On the other hand there are cases where significant difference between number of labels and ESDM exists, meaning relative ratio even higher than 200% as shown in Table III. These examples are presented in Fig.8. The more complex content produces large differences, and the errors here are due to texture related to greenary regions. This is presented by white pixels in lower right representations in Fig.8(a) and Fig.8(b), where red points are superimposed labels related to people.

The results from the second phase and the calculation of BM1 are illustrated in Fig. 9 (a)-(b). The obtained results are shown in Table III, showing the significance of semantic approach. Here, BM1 gives up to 13% relative ratio, where BM2 shows that there is possibility for firther improvements in some cases according to selection of structuring element.

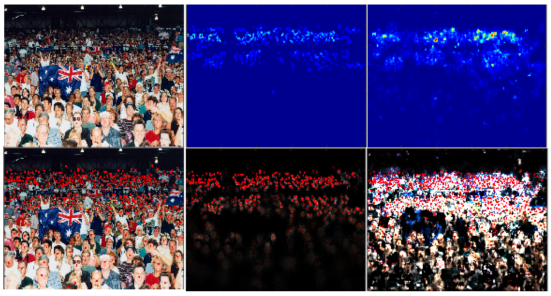

Fig. 7. Image 5 from test A representations: original (labels:431), GTDM (sum:428.6), ESDM (sum:435.1) (upper row) and the representations with labels superimposed, respectively (lower row).

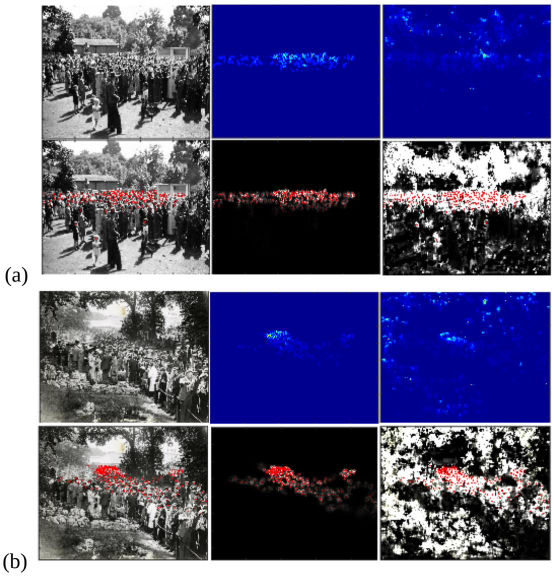

Fig. 8. The representations for: (a) image0, Atest (labels:172, GTDM-sum:171.8, ESDM-sum:598.7); (b) image3, Atest (labels:211, GTDM-sum:210.4, ESDM-sum:654.5)

The parallel networks are needed because they enable detection of objects of different sizes. Thus, three filters of sizes 3x3, 5x5 and 7x7 are applied to detect heads of lower and higher size regardless of distance from the camera being used. Higher number of parallel branches may improve results in some cases having in mind the spatial resolution of an image or object of interest.

The networks merged in a parallel architecture may have their own goals in order to deal with intra-class detections (classes of individuals with similar object sizes) and inter-class issues where the background details resemble objects of interest, i.e. individuals. The parallel structures may be even considered to set goals to deal with background false detections [16], and to train what should not belong to a foreground.

The practical purpose of such parallel networks is not only to make more accurate detections for surveillance and similar tasks, but also to further exploit the parallelization strategy. Namely, if similar processes occur in different branches one may have an improved insight into system scalability and dependency between minor challenges representing the current limitations. Here, model-parallelism is applied where the similar structure is applied with training on the same data. Data-parallelism is also an available solution, where one can analyze subsets of big data across different branches or channels [17]. Computing performance improvement and memory distribution are also possible in communications using scalable architectures and different channels.

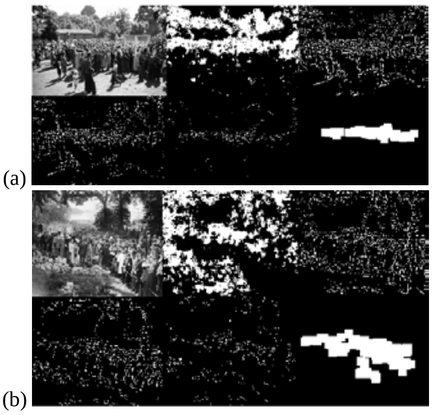

Fig. 9. Input, ESDM mask, median residual, extreme removal, local representation, and morphological result for: (a) image0 and (b) image3.

TABLE III
RELATIVE RATIO RESULTS FOR DIFFERENT BINARY MASKS

Binary mask (BM)	image0 - est. num. (R [%])	image3 - est. num. (R [%])
Ground BM	171.8 (-)	210.4 (-)
Estimated BM	598.7 (248.5%)	654.5 (211.1%)
Proposed BM1	**184.2 (7.2%)**	**183.7 (12.7%)**
Proposed BM2	215.2 (25.3%)	**212.3 (0.9%)**

V. CONCLUSION

CNN as a concept has great potential to obtain automatic estimation of the people number in a crowd. Here, a parallel network architecture with post-processing step is used for detection task. The binary mask (BM1) based approach shows the possibility to refine results, or to even show when the results of CNN is not satisfying.

Future analysis should be made towards contextual deep learning solutions related to more semantic segmentation based methods. More complex architecture could be made for making accurate binary mask estimations, where the concept of parallelism may provide scalable solutions with improved computing performance and memory distribution.

Acknowledgment

This research is supported and partially funded by Serbian Ministry of Education, Science and Technological Development.

References

[1] J. Weiss, “How reports can estimate the number of people in a crowd,” Int. Journalists' Network, 2013. https://ijnet.org/en/story/how-reporters-can-estimate-number-people-crowd (*last accessed* 12.07.2020.)

[2] Million Man March, https://www.britannica.com/event/Million-Man-March (*last accessed* 12.07.2020.)

[3] S. Ali, and M. Shah, “A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-6, 2007.

[4] K. Chen, C.C. Loy, S. Gong, and T. Xiang, “Feature mining for localised crowd counting,” in *Proc. Brit. Mach. Vis. Conf.* - BMVC, vol. 1, no. 2, p. 1-3, Sept. 2012.

[5] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, “Deep people counting in extremely dense crowds,” in *Proc. of the 23rd ACM international conference on Multimedia,* pp. 1299-1302, 2015.

[6] Y. Zhang, Z. Desen C. Siqin, G. Shenghua Gao, and M. Yi, “Single-image crowd counting via multi-column convolutional neural network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 589-597, 2016.

[7] S. Huang, X. Li, Z. Zhang, F. Wu, S. Gao, R. Ji, and J. Han, “Body structure aware deep crowd counting,” *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp.1049-1059, 2017.

[8] W. Liu, M. Salzmann, and P. Fua, “Context-aware crowd counting,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition* - CVPR, pp. 5099-5108, 2019.

[9] S. Jiang, et. al, “Mask-aware networks for crowd counting,” *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 1-10, 2019.

[10] H. Bai, S. Wen, and S.-H. G. Chan, “Crowd counting on images with scale variation and isolated clusters,” in *Proc. of the IEEE International Conference on Computer Vision Workshops* - ICCVW, pp. 1-10. 2019.

[11] J. Wan and A. Chan, “Adaptive density map generation for crowd counting,” in, pp. 1130–1139, 2019. in *Proc. of the IEEE Int. Conf. on Computer Vision* - ICCV , pp. 1130-1139, 2019.

[12] Pytorch, https://pytorch.org/ (*last accessed* 03.06.2020.)

[13] Visdom https://pypi.org/project/visdom/0.1.7/ (*last accessed* 03.06.2020.)

[14] R.C. Gonzalez, and R.E. Woods, *Digital image processing*, 2007.

[15] Sample https://stampaday.wordpress.com/2017/10/13/the-death-of-king-bhumibol-thailand-post-special-sheet-2016/ (*last accessed* 12.07.2020.)

[16] B.Y. Lin and C.S. Chen, “Two parallel deep convolutional neural networks for pedestrian detection,” In *Proc. of the IEEE International Conference on Image and Vision Computing New Zealand* (*IVCNZ*), pp. 1-6, November, 2015.

[17] S. Lee, D. Jha, A. Agrawal, A. Choudhary, and W.K. Liao, “Parallel deep convolutional neural network training by exploiting the overlapping of computation and communication,” In *Proc. of the 24th IEEE International Conference on High Performance Computing* (*HiPC*), pp. 183-192, December, 2017.