Crowd Counting Mapping to Make a Decision

Enas Faisal, Azzam Sleit
Computer Science Department
The University of Jordan
Amman-Jordan

Rizik Alsayyed
Business Information Technology
The University of Jordan
Amman-Jordan

***Abstract*—Congestion typically occurs when the number of crowds exceeds the capacity of facilities. In some cases, when buildings have to be evacuated, people might be trapped in congestion and cannot escape from the building early enough which might even lead to stampedes. Crowd Congestion Mapping (CCM) is a system that enables organizations to find information about the crowd congestion in target places. This project provides the ability to make the right decision to determine the reasons that led to that and to do the appropriate procedures to avoid this from happening again by optimizing locations and dimensions of the emergency exits less congested path on the target places. The system collects crowd congestion data from the locations and makes it available to corporations via target map. The congestion is plotted on target place map, for example, the red line for highly congested location, the pink line for mildly congested location and green line for free flow of humans in the location.**

Keywords*—*Crowd; map; image processing; human detection; threshold; recognition

I. Introduction

There is great interest in the surveillance system, especially in cases of high congestion as the ability to identify objects and follow them or to detect the times that become congestion. These things have become important but difficult to measure, so there is concentrate on computer vision algorithms to try to solve these problems.

Work on the crowd has gained much attention in recent years for a variety of applications such as video surveillance, public safety design and traffic control. Many researchers have handled more than one aspect of the analysis of crowded scenes such as counting [9], [10], detecting anomalies [12], Segmentation [11], and many other aspects [13]-[15], [19], [20].

To help institutions and centers make appropriate decisions in the event of congestion in advance, a system for that is needed. In this paper, a system was proposed for the identification of cases of crowded gatherings that may occur at certain times, which may lead to security problems, interruptions or other problems.this system is based on image feature extraction and Feed-Forward Neural Networks (FFNN). This system plotted gathered information on the map to explore the hazardous locations to take a decision in this situation.

The rest of the paper is organized as follows. Section II discusses the related work, The proposed Algorithm is explained in Section III, Section IV discusses and analyzes the results. Finally, the conclusions are drawn in Section V.

II. Related Work

In the field of computer vision, many motions detection algorithms are introduced in literature to detect abnormal movement in dynamic congestion, but building an automatic detection system is still a challenge.

Brostow and Cipolla in [4] present a system to detect specific people in congestion, but this system has encountered problems when have a noise or objects exist in congestion such as kiosk or antiques.

Also, Pathan et al. [5] worked out a system in which they identified all erroneous movements during the congestion stages. But this system does not produce good results when there is high congestion where it is not possible to recognize the number of people because they use subtractions process in the detection people.

Moreover, Krausz and Bauckhage in [2] proposed a system works automatically for identifying the critical situation during an increased number of the public by using system alarms, but this system showed error in the detection even at normal state. Dee and Caplier in [6] suggest a system depend on representing the movement pattern of the crowd.

Shang et al. [10] proposed an end-to-end model using CNN to handle the process of counting in crowd images by simultaneously learning local and global count on the full sized input images. Onoro-Rubio and Lopez-Sastre in [16] proposed Hydra CNN to addressed the scale issue by proposing a scale aware counting model by uses a pyramid of image patches extracted at multiple scales to perform the final density prediction. Zhang et al. [17] proposed Multi-column Convolutional Neural Network (MCNN) architecture to extract features generated by filters of different scales to generate the final prediction for its crowd density map.

Boominathan et al. in [18] proposed a novel deep learning framework by a combination of deep and shallow to tackle the issue of scale variation for estimating crowd density to predict the density map for a given crowd image.

Rohit et al. in [21] they proposed epsilon Support Value Regression (SVR) fusion-based approach to help detect and sort out people in images of the highly blocked crowd by influence information on the global construction of the crowd scenes and identifying people in these scenes.

III. Proposed Method

First of all, we acquired the video feed from surveillance cameras of four locations in City Mall in Amman city. In order to detect from the video, we applied various images processing procedures. The extracted image frames were divided into several blocks. Then analyze each block by finding the circular or the semicircular shape of edge detection.

The result was analyzed the collected objects based on bright and dark color. Then the objects were counted and people density can be plotted on map. The architecture of the proposed system is presented via flow diagram as shown in Fig. 1.

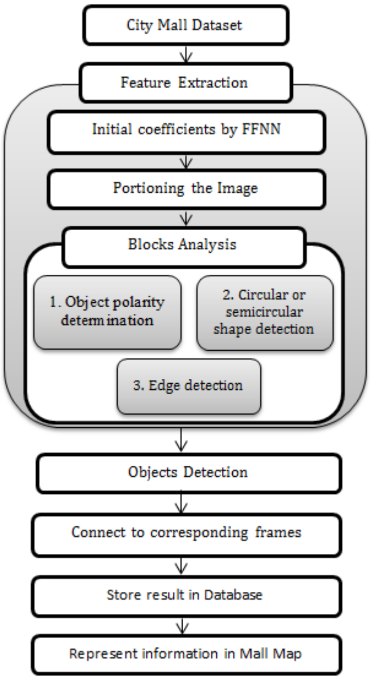

City Mall Dataset
Feature Extraction
Initial coefficients by FFNN
Portioning the Image
Blocks Analysis
1. Object polarity determination
2. Circular or semicircular shape detection
3. Edge detection
Objects Detection
Connect to corresponding frames
Store result in Database
Represent information in Mall Map

Fig. 1. Proposed algorithm architecture.

A. Feature Extraction

1) Initial coefficients by FFNN: We use Feed Forward Neural Network (FFNN), which contains a set of layers, the first layer is defined the variables of the network, and the last layer is the output layer. Between these two layers are called hidden layers. We use FFNN as a way, instead of randomly determining the initial coefficients (see Fig. 2), where:
Rmin = Minimum radius of the interested object (human head).
Rmax = Maximum radius of the interested object (human head).
EdgTh = Edge detection threshold.
Sens = Sensitivity level for interested object detection.

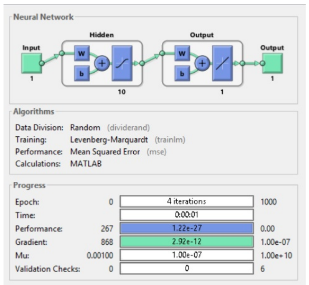
Therefore, instead of having the initial coefficients randomly and changing their values by trial and error. We use FFNN to accurately measure them, based on our knowledge about the accurate number of persons in the image. So any new test image can be analyzed more accurately.

2) Partitioning the image: A given a training video frame of City Mall for Ground floor i, where i = 1, 2...N and N denotes the total number of training frames, we first partition the frame into K blocks regions (see Fig. 3). The frame is dividing into 8 blocks and the coordinates of each block by (1), (2):

$$BL = FL/8 \tag{1}$$

$$BW = FW/8 \tag{2}$$

Where BL is the block length, BW is the block width, FL is frame length, and FW is a frame width.



Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MATLAB

Progress

Epoch:	0	4 iterations	1000
Time:		0:00:01	
Performance:	267	1.22e-27	0.00
Gradient:	868	2.92e-12	1.00e-07
Mu:	0.00100	1.00e-07	1.00e+10
Validation Checks:	0	0	6

Fig. 2. ANN configuration.

B. Blocks Analysis

For every blocks region we apply the object polarity determination then we find the circular and the semicircular shapes after that we apply edge detection to reduce the false positive.

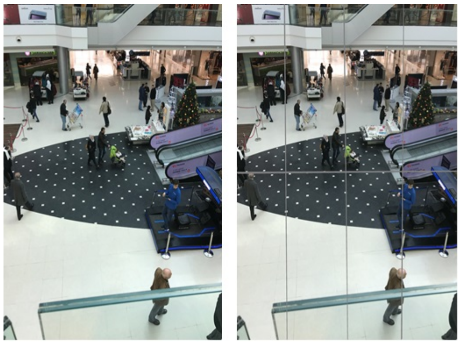


Fig. 3. Image before and after blocking.

1) Object polarity determination: In this process we detect the color by the polarity of the image horizontally, and determine the real bright and real dark objects, by (3) and (4) below:

$$\frac{\sum_{21}^{c-21}(\sum_{-20}^{20}\sum_{1}^{c}\sum_{1}^{r}(\left|p(r-1_{-p(r)})\right|)>20)}{41} \tag{3}$$

$$if(Eq(3)) < \frac{(\sum_{1}^{c}\sum_{1}^{r}(|p(r-1)-p(r))|)>20)}{c} \tag{4}$$

Where c is the column index, r is the row index and p is the current pixel, this equation investigates the pixels for each

41 column and compares them with the next 42 columns. This way we can determine the polarity of the image horizontally, and determine the real bright and real dark objects.

2) Circular or semicircular object detection: After portioning the frames for blocks and applying the object polarity, then we will find the circular and semicircular shape by using Hough Circle transform. Hough transform is used to determine which circuits are present in an image given, whether the circuit is a wheel, head or other[1]. In this paper, we use the circular Hough transform (CHT) [7] and randomized Hough transforms [8]. How the CHT works is based on (5):

$$(x-a)^2+(y-b)^2=r^2 \tag{5}$$

Where a and b are the coordinates of the circle center, and r is its radius. In Fig. 4 the solid line represents the circle and the dashed lines represent the hypothetical circles from the points on the edge of the actual circle.

The hypothetical circles in the image space can be presented in a parameter space as a circular cone. If r is increased and equal to r, the hypothetical circles will intersect at point c, and a circle can thereby be detected. The randomized Hough transform is also used to detect ellipses. The ellipse detection method requires five parameters: a center coordinates (x_0, y_0), a rotation angles α, and the half-length of the major and minor axes, a and b, respectively. These parameters are defined by the equations below:

$$x_0=\frac{x_2+x_1}{2}, y_0=\frac{y_2+y_1}{2} \tag{6}$$

$$a=\frac{\sqrt{(x_2+x_1)^2+(y_2+y_1)^2}}{2}, b^2=\frac{a^2d^2sin^2T}{a^2-d^2cos^2T} \tag{7}$$

$$\alpha=arctan\frac{y_2-y_1}{x_2-x_1} \tag{8}$$

Calculate the parameters in (6) across ellipse geometry across (7). Here the coordination (x_1, y_1) and (x_2, y_2) are endpoints of maximizing axis, and these equations are repeated on all pixels in the image to determine the ellipse.

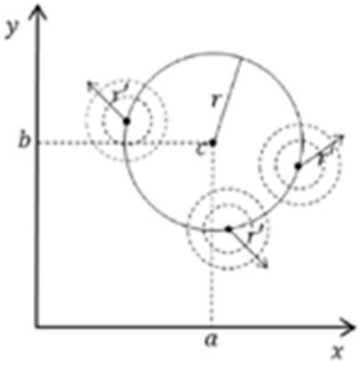


Fig. 4. The Hough transforms [1].

3) Edge detection: After finding circular and semicircular shapes we apply the Canny Edge Detection to reduce false positive. The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images [3]. We apply the process of canny edge detection algorithm at the output images from the previous phase, as shown in Fig. 5. With this phase, we finish the feature extraction process and final result appears in Fig. 6.

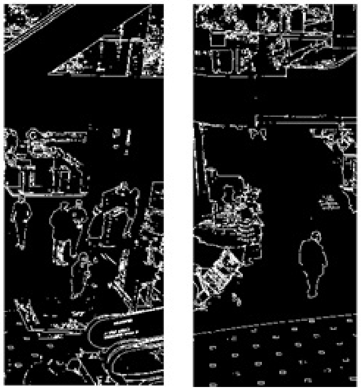


Fig. 5. Some block after edge detection.

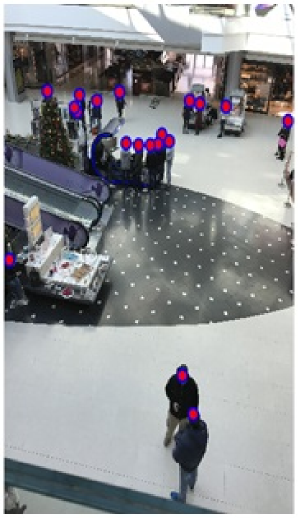


Fig. 6. The final result of crowd detection.

C. Objects Detection

As soon as we determine the location of the objects of interest, the algorithm itself returns the coordinates of all the detected objects and stores them in the vector. These coordinates we use them in "plot" command to put circles over them. Moreover, the number of the elements inside this vector equals the number of interesting objects.

D. Connect to Corresponding Frames

After counting the number of objects in each frame, a link is made between each frame with the appropriate location in the mall. This algorithm applies to the four locations on the ground floor of the mall. After that the number of objects in each frame is collected for each location to determine the place where it is most crowded to apply a set of procedures to mitigate them in the future.

E. Thresholding

The process of determining the location that has the highest percentage of crowded depends on the threshold, which is based on the capacity of the area, as defined in (9):

$$Areacapacity = Imagearea/personoccupationarea \quad (9)$$

At the threshold algorithm as shown in Algorithm 1, we have assumed arbitrary values to calculate the threshold, since we have excluded half of the location area as an arbitrary value, assuming that humans are not present there, such as kiosks, decoration, stairs, etc. Thus, we assume that considered the threshold for the other half as the place of the capability of human existence.

As shown in Algorithm 1: line 1 calculate the area of the location as C and in line 2 divide the area by two to calculate the Ca as the area where humans are able to exist. In line 3-4, threshold Th computes as the human area Ca divided by average human area as ε.

In line 5-10: threshold Th used to decide the crowded locations by comparing the value of Th with φ, if φ value more or equal to Th value then this location has highly crowded and this location needs procedures to mitigate the hazardous companies with congestion, but if φ value between the half of Th value and Th-1 then this location has medium crowded, the location considered not crowded if φ value less than Th-1.

φ= the AVG detected # of objects $\forall$ frames in one Location.

Algorithm 1: Threshold Algorithm

```
C = Full image area
Compute Ca            */where Ca = C / 2
$\varepsilon$ = Avg human area
Th = Ca / $\varepsilon$
If $\varphi >$ = Th Then
Highly Crowd
Elseif $\varphi$ >= Th /2 and $\varphi$ ¡= Th-1 Then
Medium Crowd
Elseif $\varphi <$ = Th-1 Then
Not Crowd
```

F. Plotting Information on Map

The information of object density of particular location in City Mall was plotted on City Mall map, where the Green color indicated to free flow of people, red line indicated high crowd and pink line indicated mild crowd as shown in Fig. 7.

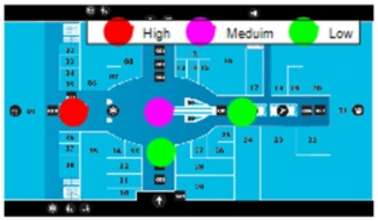


Fig. 7. Map after colored.

IV. Results and Discussion

The experiments conducted over MATLAB environment. The images it has been taken from various cameras of four locations placed at different angles on the ground floor of City Mall (Amman city).

The evaluation of the effectiveness of the proposed method for the four locations has been tested as part of the study. As the given data in Table I, the number of humans detected by the system was compared with the actual number of humans in image frames. The result was 96% accurate for the camera at the location 4 and 99%-98% accurate for the camera on the other three locations.

TABLE I. Detection Rates for Each Location

City Mall Locations	Average Performance
Location 1	98%
Location 2	98%
Location 3	99%
Location 4	96%

The dataset for each location is visualizing the real number of crowds in each image frame against the measured number of proposed algorithm as shown in Fig. 8.

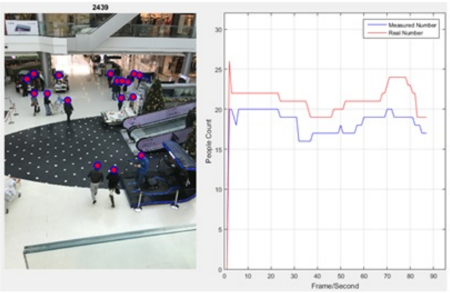


Fig. 8. Rate of Measured number vs. Real number.

The cent percent accuracy was not obtained because of various reasons. When the brightness is low this approach has a problem to differentiate between the human head and objects. And when their dark background with a person wraps his back to the camera (his face reversing the camera) made a difficulty to distinguish between the color of the hair and the background during the process of polarity. We expect that illumination might help in solving this problem at future work.

The other problem that we encountered is the people who stand behind the object (such as kiosks, etc.) wherein the visible eye there is difficulty in distinguishing the presence of that person or not.

The location of the camera affects the size of the head radius for people are close or far from the camera, which creates a problem in the training process of Neural network, because we give a range of radius between the largest and smallest value, and therefore may affect the detection of humans with object have radius matched with given range (such as the wheel of the kiosk).

We recommend using strong lighting and do not use a dark background to be bright as much as they can. Make coverage radius of the camera for human heads within a small range, therefore no big gap within the same range. The process of calculating the crowd ratio depends on the environment since the environment is determined the level of the crowd as it varies between games at the stadium or in a mall. In this work we assumed a congestion ratio based on the mall environment.

V. Conclusion

Overcrowd occur in institutions and malls in excess of their ability to withstand these numbers, especially in times of holidays and other occasions. In these times the crowd is getting so critical, leading to problems in the stampede among the crowd. In this paper, crowd counting mapping approach was proposed, which based on image feature extraction and FFNN. In this method, high congestion was detected early to make the appropriate decision to deal with this situation so that no problems occur in the future. After testing, high detection rates were achieved, their accuracy could be as high as 98%. The proposed algorithm uses a powerful set of features which proved to be effective in the crowd detection and we believe that our approach would be integrated with any organization surveillance system. In the future we want to improve neural network training using one of the meta-heuristic algorithms to give more accuracy in detecting people.

References

[1] Young-Jin Cha , Kisung You , Wooram Choi ,Vision-based detection of loosened bolts using the Hough transform and support vector machines, Elsevier, 2016.

[2] B. Krausz and C. Bauckhage, Automatic detection of dangerous motion behavior in human crowds, in Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on, pp. 224229,2011.

[3] Paul Bao, Lei Zhang, and Xiaolin Wu, Canny Edge Detection Enhancement by Scale Multiplication, IEEE transactions on pattern analysis and machine intelligence, VOL. 27, NO. 9, SEPTEMBER 2005.

[4] G. Brostow and R. Cipolla, Unsupervised bayesian detection of independent motion in crowds, in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 1, pp. 594601, 2006.

[5] S. Pathan, A. Al-Hamadi, and B. Michaelis, Crowd behavior detection by statistical modeling of motion patterns, in Soft Computing and Pattern Recognition (SoCPaR), 2010 International Conference of, pp. 8186, 2010.

[6] H. Dee and A. Caplier, Crowd behaviour analysis using histograms of motion direction, in Image Processing (ICIP), 2010 17th IEEE International Conference on, pp. 15451548, 2010.

[7] H.K. Yuen, et al., Comparative study of Hough transform methods for circle finding, Image Vis. Comput. 8 (1) (1990) 7177.

[8] C. Basa, M. Talo, R. Brad, Randomized Hough transform for ellipse detection with result clustering. Computer as a Tool, 2005, EUROCON 2005. The International Conference on. Vol. 2, IEEE, 2005.

[9] A. B. Chan and N. Vasconcelos. Counting people with lowlevel features and bayesian regression. IEEE Transactions on Image Processing, 21(4):21602177, 2012.

[10] C. Shang, H. Ai, and B. Bai. End-to-end crowd counting via joint learning local and global count. In IEEE ICIP, pages 12151219. IEEE, 2016.

[11] K. Kang and X. Wang. Fully convolutional neural networks for crowd segmentation. arXiv preprint arXiv:1411.4464, 2014.

[12] H. Rabiee, J. Haddadnia, H. Mousavi, M. Kalantarzadeh, M. Nabi, and V. Murino. Novel dataset for fine-grained abnormal behavior understanding in crowd. In IEEE International Conference on AVSS, pages 95101. IEEE, 2016.

[13] J. Shao, C. Change Loy, and X. Wang. Scene-independent group profiling in crowd. In IEEE CVPR, pages 22192226, 2014.

[14] J. Shao, K. Kang, C. C. Loy, and X. Wang. Deeply learned attributes for crowded scene understanding. In Proceedings of the IEEE CVPR, pages 46574666. IEEE, 2015.

[15] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert. Density-aware person detection and tracking in crowds. In IEEE ICCV, pages 24232430. IEEE, 2011.

[16] D. Onoro-Rubio and R. J. Lopez-Sastre. Towards perspective-free object counting with deep learning. In ECCV, pages 615629. Springer, 2016.

[17] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Singleimage crowd counting via multi-column convolutional neural network. In Proceedings of the IEEE CVPR, pages 589 597, 2016.

[18] 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 Conference, pages 640644. ACM, 2016.

[19] V. Sindagi and V. Patel. CNN-based Cascaded Multi-task Learning of High-level Prior and Density Estimation for Crowd Counting. (AVSS) , IEEE, 2017.

[20] V. Sindagia, V. Patelb. A Survey of Recent Advances in CNN-based Single Image Crowd Counting and Density Estimation. Elsevier, 2017.

[21] Rohit, V. Chauhan, S. Kumar, S. K. Singh .Human Count Estimation in High Density Crowd Images and Videos. Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), IEEE, 2016.