Crowd Abnormal Behaviour Patterns: Survey and Detection

Stanislav V. Sholtanyuk

***Abstract*—** Nowadays, crowd research and solving various tasks regarding to crowd's motion and behavior (e.g. crowd characteristics estimation, its motion and behavior prediction, etc.) is one of the well-studied, yet promising areas in computer vision and data analysis. Research of crowd motion and behavior in emergencies, like catastrophes, accidents, and panicking, are of special interest. In such situations, it's crucial to predict crowd's motion and behavior quite accurately so immediate decisions and actions could be done.

In this paper, various patterns of crowd behavior, regular and abnormal, are considered. These patterns are designated by crowd scene structures and motion patterns (lanes motion, leader-following motion, aggregation, dispersion, etc.) which are detected or predicted by means of computer vision, including optical flow and convolutional neural networks (CNNs), CSRNet, FlowNet are among them. CSRNet and its modifications were used for building density maps. FlowNet architecture was considered for optical flow forecasting.

***Keywords*—** computer vision; convolutional neural networks; crowd behavior; crowd motion; optical flow.

I. Introduction

Nowadays problems connected with abnormalities in crowd motion and behavior arise more often and often. Moreover, those abnormalities must be detected and dealt with urgently to prevent any casualties. The problem became casual for many spheres of life, including traffics, surveillance systems, defense, security, building layout design, among many others.

Crowd abnormal behavior problem is manyfold. At first, many patterns can arise in different locations and situations, each of them must be addressed on its own way. Those situations can differ by their development speed, emergency, number of actors involved, and many other aspect. Besides, those situations can arise simultaneously in different observed places, and making decision about which of the incidents must be addressed first is crucial.

In this paper, we propose a model based on two artificial neural networks (ANN). One of them, FlowNet [1], estimates crowd motion using the optical flow concept, and the other one, based on CSRNet [2], measures crowd density and count for a given image. After video processing by these ANNs, we obtain two frame sequences which are segmented, and frames of interest are chosen. By using both ANNs, we can effectively segment separate frames on clusters by their size and motion direction.

The rest of the paper is organized as follows. In Section 2, we discuss the related works, and give the brief survey of crowd behavior patterns. Section 3 provides the methodology for crowd abnormal behavior detection using CSRNet and Flownet. In Section 4, some examples of results are given. Finally, conclusion is presented in Section 5.

II. Related Works and The Problem Formulation

To perform crowd abnormal behavior classification, detection, and prediction, variety of different approaches and methods have been developed and proposed.

Using ANNs is arguably the most popular approach in crowd images and videos research. For example, crowd counting can be effectively performed by using ANNs. Zhang et al. [3] proposed a CNN to estimate both crowd count and crowd density on a crowded scene. The CNN is trained on patches randomly selected from training images. As a result, a density map

S. V. Sholtanyuk Belarusian State University, Minsk, Belarus (e-mail: ssholtanyuk@bsu.by).

of considered crowded scene is obtained. CSRNet [2] works in similar way, but it consists of several first layers of VGG-16 for feature extraction, and custom end layers which use dilated kernels to deliver larger reception fields and to replace pooling operations.

As for crowd motion and behavior detection and prediction, there are the following research where new ANN models are developed and used. Zhao et al. [4] proposed an ANN for separate human postures segmentation on an image with a low-crowded scene. The ANN consists of three connected Generative Adversarial Networks (GAN) addressing their own subtasks (extracting the most noticeable postures silhouettes from an image, parsing those silhouettes into separate instances like body parts and clothing, and clustering the instances into separate human postures) which are easier than the original problem. Zhang et al. [5] developed the SR-LSTM net to predict pedestrians' walking trajectory on a given video. For that, they consider a scene and pedestrians on it as a fully connected graph so it can be used as a variant of Graph Convolution Network [6, 7]. Zhao et al. [8] developed and used the Multi-Agent Tensor Fusion (MATF) encoder and the decoder architecture for trajectory prediction of vehicles and pedestrians. In that research, they consider the spatial structure of scenes and actors rather than just their positioning so the problem of modeling spatial relationships between actors and obstacles, as well as actors interacting with each other, has arose. To model those interactions, they used convolutional layers of MATF. Liu, Yan, and Alahi [9] proposed a model called Social NCE to forecast "rare but dangerous" scenarios like collisions between pedestrians. For that, they collected "negative" data to train the model and proposed a social sampling strategy to address number of problems including considering socially aware behaviors and extreme actors' motion cases. Moreover, in their resent work [10] they addressed the problem of reusability training data and models for new environments by using the causal invariance and its implementation in a novel modular (rather than dense) architecture.

One of other approaches relies on using optical flow. Andrade, Blunsden, and Fisher [11] proposed a framework for detection events and emergencies in crowds. For this, they used hidden Markov models been trained on simulated dense crowd data [12]. Ryan et al. [13] considered structures which they called textures of optical flow to detect abnormalities on crowded scenes. That approach allows detect abnormal objects like bicycles and vehicles, as well as optical flow uniformity violations, thus motion abnormalities. Singh and Singh [14] also used optical flow-based method considering crowd escape behavior as an indicator of abnormal event. Chen et al. [15, 16] proposed the concept of integral optical flow. Unlike the classical optical flow, which estimates pixel displacements between two neighboring frames of a video, an integral optical flow "is the intuitive idea of accumulation of optical flow for several consecutive frames", which can mitigate the problem of distinction between moving scene background and the actors of one's interest.

In these and other researches, their authors considered various abnormal behavior patterns. They can be divided into the following categories:

- By motion type. The most common patterns researched and described by various researchers include aggregation and dispersion [15, 16], lanes motion [17], queuing [18]. A comprehensive survey of crowd motion types can be found at [19].
- By crowd's behavior, e.g. chaotic [20], leader following [21, 22], self-organizing [17, 18] crowds.
- By emergency factor: no emergency, escaping from danger [14], evacuation [22, 23].

In this paper, we considered non-emergency line-directed crowd motions. Motions having opposite or significantly different direction compared to the averaged motion are considered as abnormal ones.

III. METHODOLOGY

A. CSRNet

In this research, we used CSRNet shallowed modification [24] which has been proved to estimate crowd counts successfully without wasting too much time. The original CSRNet consists of 13 frontend layers derived from VGG-16, and 5 convolutional layers using dilated kernels with dilation rate 2 (Fig. 1). The shallowed modification contains 9 frontend layers and 3 convolutional layers with dilated kernels. The architectures of both original and modified networks are given on Fig. 2.

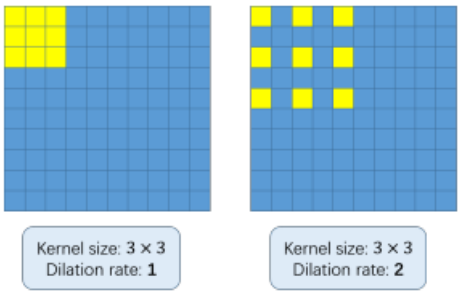


Fig. 1 3 × 3 convolution kernels with dilation rates 1 (standard kernel) and 2

	Input	array (size: 3×□×□)
1	ConvolutionLayer	array (size: 64×□×□)
2	Ramp	array (size: 64×□×□)
3	ConvolutionLayer	array (size: 64×□×□)
4	Ramp	array (size: 64×□×□)
5	PoolingLayer	array (size: 64×□×□)
6	ConvolutionLayer	array (size: 128×□×□)
7	Ramp	array (size: 128×□×□)
8	ConvolutionLayer	array (size: 128×□×□)
9	Ramp	array (size: 128×□×□)
10	PoolingLayer	array (size: 128×□×□)
11	ConvolutionLayer	array (size: 256×□×□)
12	Ramp	array (size: 256×□×□)
13	ConvolutionLayer	array (size: 256×□×□)
14	Ramp	array (size: 256×□×□)
15	ConvolutionLayer	array (size: 256×□×□)
16	Ramp	array (size: 256×□×□)
17	PoolingLayer	array (size: 256×□×□)
18	ConvolutionLayer	array (size: 512×□×□)
19	Ramp	array (size: 512×□×□)
20	ConvolutionLayer	array (size: 512×□×□)
21	Ramp	array (size: 512×□×□)
22	ConvolutionLayer	array (size: 512×□×□)
23	Ramp	array (size: 512×□×□)
24	ConvolutionLayer	array (size: 512×□×□)
25	Ramp	array (size: 512×□×□)
26	ConvolutionLayer	array (size: 512×□×□)
27	Ramp	array (size: 512×□×□)
28	ConvolutionLayer	array (size: 512×□×□)
29	Ramp	array (size: 512×□×□)
30	ConvolutionLayer	array (size: 256×□×□)
31	Ramp	array (size: 256×□×□)
32	ConvolutionLayer	array (size: 128×□×□)
33	Ramp	array (size: 128×□×□)
34	ConvolutionLayer	array (size: 64×□×□)
35	Ramp	array (size: 64×□×□)
36	ConvolutionLayer	array (size: 1×□×□)
	Output	array (size: 1×□×□)

a

	Input	array (size: 3×□×□)
1	ConvolutionLayer	array (size: 64×□×□)
2	Ramp	array (size: 64×□×□)
3	PoolingLayer	array (size: 64×□×□)
4	ConvolutionLayer	array (size: 64×□×□)
5	Ramp	array (size: 64×□×□)
6	PoolingLayer	array (size: 64×□×□)
7	ConvolutionLayer	array (size: 128×□×□)
8	Ramp	array (size: 128×□×□)
9	ConvolutionLayer	array (size: 128×□×□)
10	Ramp	array (size: 128×□×□)
11	PoolingLayer	array (size: 128×□×□)
12	ConvolutionLayer	array (size: 256×□×□)
13	Ramp	array (size: 256×□×□)
14	ConvolutionLayer	array (size: 256×□×□)
15	Ramp	array (size: 256×□×□)
16	ConvolutionLayer	array (size: 256×□×□)
17	Ramp	array (size: 256×□×□)
18	ConvolutionLayer	array (size: 256×□×□)
19	Ramp	array (size: 256×□×□)
20	ConvolutionLayer	array (size: 128×□×□)
21	Ramp	array (size: 128×□×□)
22	ConvolutionLayer	array (size: 64×□×□)
23	Ramp	array (size: 64×□×□)
24	ConvolutionLayer	array (size: 1×□×□)
	Output	array (size: 1×□×□)

b

Fig. 2 Architectures of two crowd counting ANNs: (a) the original CSRNet and (b) the shallowed one. "Ramp" stands for the standard ReLU activation function.

Both ANNs receive an RGB crowd image as the input and give the density map as the result. Because of the convolutions taking place during ANN processing, density map is the 2D-array having dimensions eight times smaller than the initial crowd image. An element on the cross of given row and column indicates the estimated density for the corresponding area of the original image. The density map can be pictured as a new image (Fig. 3).

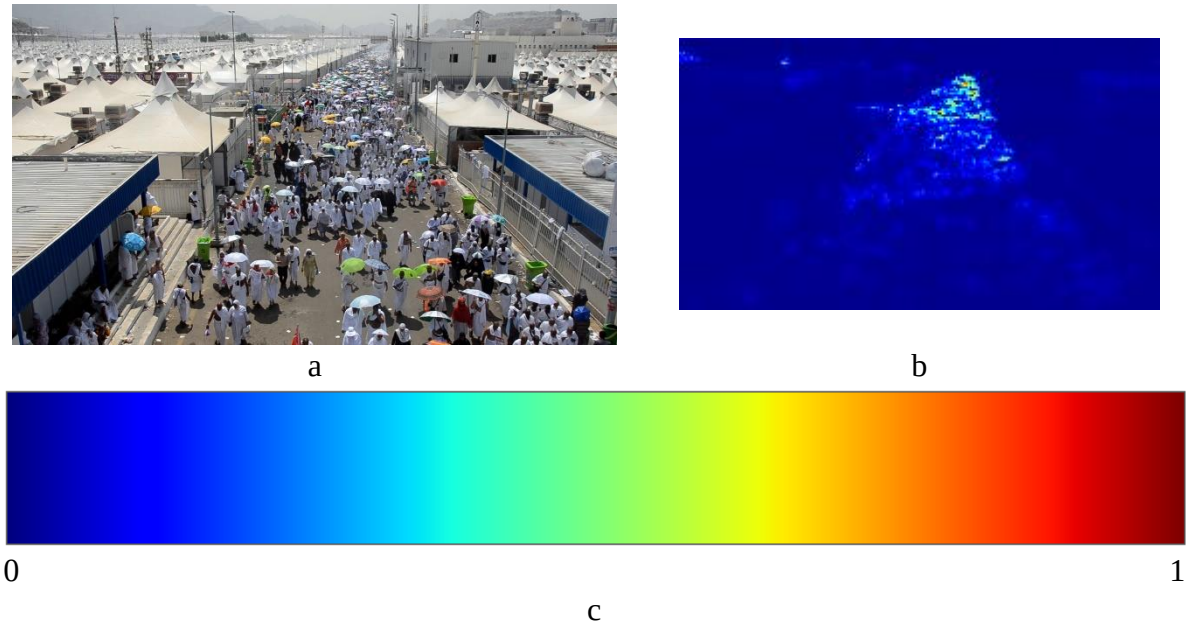


Fig. 3 (a) Original image and (b) the density map for it. Color scale (c) indicates the mapping between colors and numeric values.

The shallow network has been trained on ShanghaiTech dataset [25] of crowd images with different resolution. This dataset has been chosen because it is not difficult to obtain ground truth density maps for images from it, and these images have been collected from various sources. This dataset is divided into two parts, A and B. In the A part, there are as many images as 300 for training (Train A dataset) and 182 for testing (Test A). In the B part, there are 400 and 316 images in training and testing parts, respectively (Train B and Test B). Histograms of crowd counting on these dataset images are given on Fig. 4.

B. FlowNet

FlowNet 2.0 has been used for optical flow estimation. We decided to use its original architecture and to train it on the FlyingChairs dataset [26]. FlowNet 2.0 takes two neighbour video frames as its input and gives optical flow map as the result (Fig. 5). Every pixel on the optical flow map gets its HSB code depending on what direction and what speed it has according to the FlowNet (Fig. 6). Hue indicates the direction of a pixel motion (e.g., red for moving right, yellow is for moving to bottom, etc.), saturation indicates the speed of a pixel, and brightness of a pixel is always 1, hence wasn't considered in the research.

White color which has zero saturation stands for an immobile pixel. For a video with moving crowd captured by a fixed camera, white space is likely to represent a background. Never-the-less, one must take into consideration there can be spaces with low but non-zero saturation, hence with significant hue so it can be incorrectly interpreted as a certain motion direction.

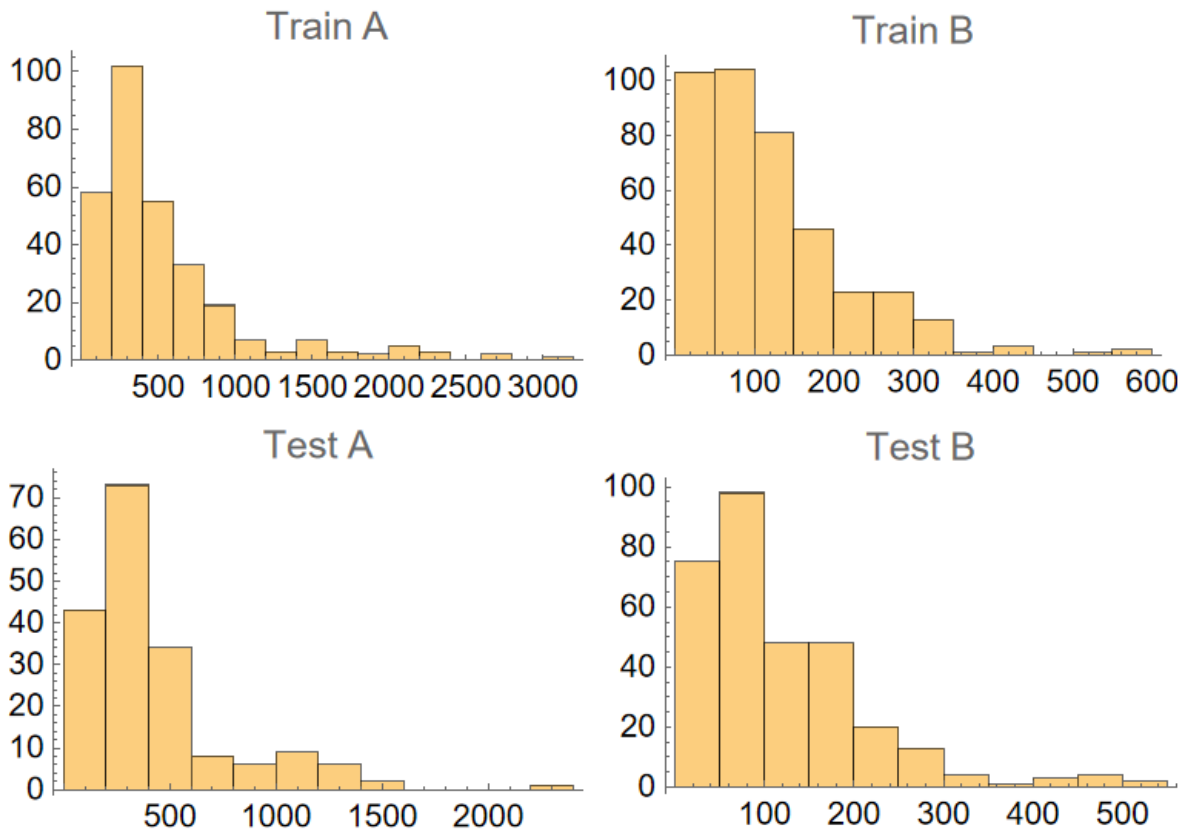


Fig. 4 Histograms for crowd counting on images from ShanghaiTech dataset. The horizontal axis corresponds to crowd counts, and the vertical axis indicates the number of images having given crowd count on them.

a b

Fig. 5 (a) Video frame and (b) the corresponding optical flow between this frame and the next one

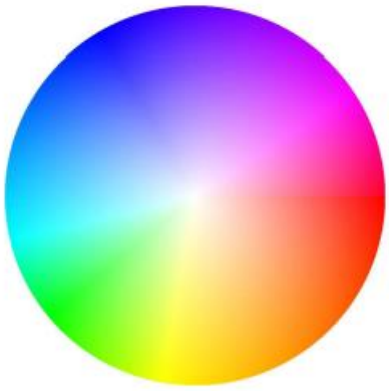


Fig. 6 Colors being used in optical flow images

C. *Video segmentation*

After obtaining density and optical flow maps for each frame of a given video, optical flow maps sequences have been processed in the following way:

1) Whole optical flow maps sequence for a video is segmented into clusters. Euclidean distance between two images as flattened $w \times h \times 3$ arrays is used for clustering images:

$$d(I_1, I_2) = \sqrt{\sum_{x=1}^{h}\sum_{y=1}^{w}\left(d_h(h_{1xy}, h_{2xy})^2 + (s_{1xy} - s_{2xy})^2\right)},$$

where $h_{1xy} \in [0,1]$ and $s_{1xy} \in [0,1]$ denote hue and saturation of the pixel from 1st image on x^{th} row and y^{th} column, $h_{2xy} \in [0,1]$ and $s_{2xy} \in [0,1]$ stand for the same thing with 2nd image, and d_h is the function calculating the distance between two hues. For example, one can consider the following function:

$$d_h(h_1, h_2) = \begin{cases} |h_1 - h_2|, |h_1 - h_2| \le \frac{1}{2}, \\ 1 - |h_1 - h_2| \text{ otherwise.} \end{cases}$$

As it appeared, for an optical flow maps sequence the most likely situation is, there will be one major cluster among multiple minor ones. On Fig. 7, clusterization results for a couple of videos are shown. These squares denote frames sequences and should be read by rows (first row contains pixels denoting first frames in the sequence, then the second row goes, etc.). Each color (except for black) stands for a separate cluster, and black pixels are padding ones, so we obtained exact squares.

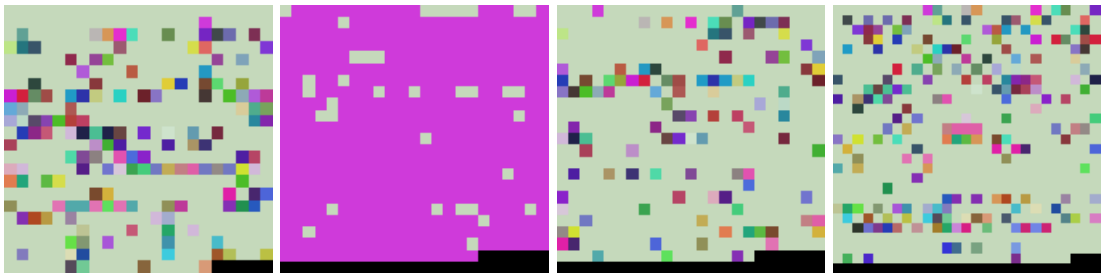


Fig. 7 Clusterization results visualization examples

2) Finding the longest subsequence of frames not lying in the major cluster.
3) Considering separate frames from the resultant subsequence.

D. *Single frame segmentation*

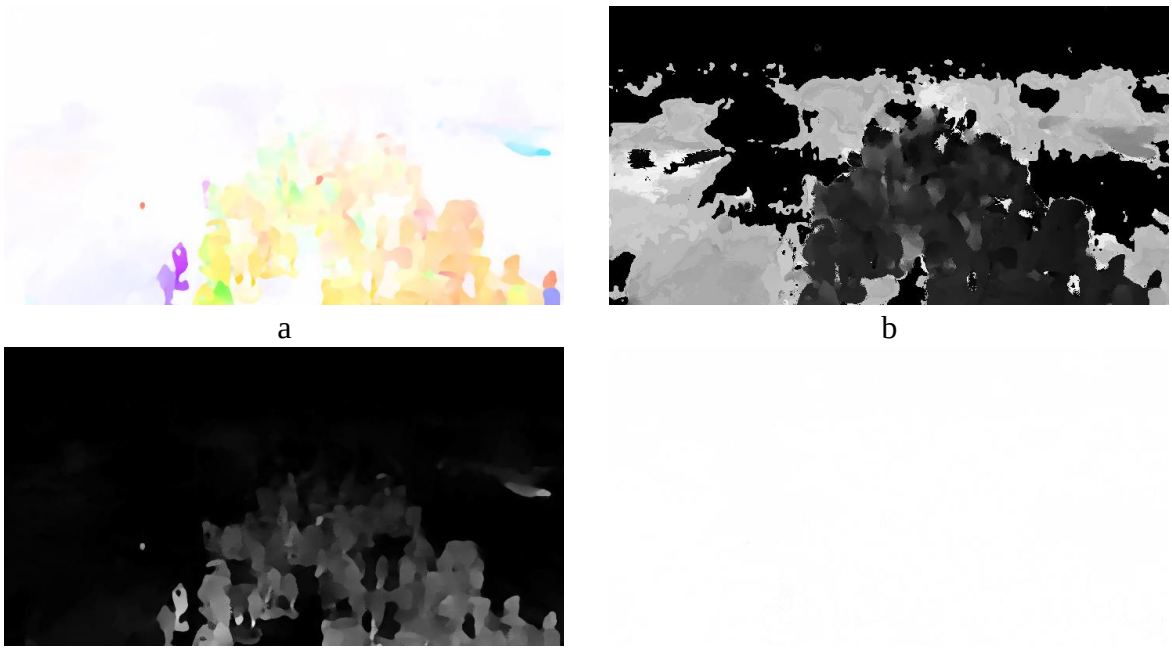
For a frame of our interest, its segmentation has been performed in the following way:

1) Frame is blurred with median filtration. Kernel with radius 6 was used (Fig. 8).

a b

Fig. 8 (a) Initial optical flow image and (b) this image after blurring

2) Blurred image separated into 3 channels of HSB color system (Fig. 9). All pixels have brightness 1, so Fig. 9d is just a blank image.

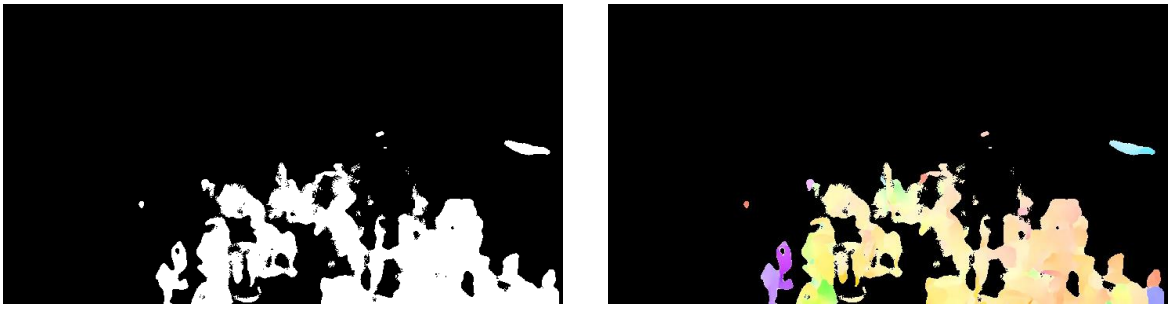


a b

c d

Fig. 9 (a) Blurred image, (b) its hue, (c) saturation, and (d) brightness channels

3) Saturation channel image is clustered into two clusters, one of them being the motionless background, and the other one representing moving actors. After segmenting the background, we can apply a mask to blurred optical flow map (Fig. 10).



a b

Fig. 10 Background extraction

4) After the background extraction, the blurred image is clusterized by watershed method. Clustering by watersheds is proved to be a fast, easy to implement and use, and effective method of image processing [27-29]. We used the gradient algorithm with a specified minimal saliency threshold. Saliency is a number in [0, 1] segment indicating the significance of a i^{th} cluster:

$$\mu_i = \frac{\max\limits_{(x,y)\in S_i} B(x,y) - \min\limits_{(x,y)\in S_i} B(x,y)}{\max\limits_{(x,y)\in I} B(x,y) - \min\limits_{(x,y)\in I} B(x,y)},$$

where S_i stands for i^{th} cluster, I is the whole image, and $B(x, y)$ is a function which matches a pixel with (x, y) coordinates to a "height" of this pixel. If a segment has saliency μ_i less than a given threshold, it merges with a neighbor segment. Hence, the larger threshold, the less clusters image is divided into.
Experiments proven any minimal saliency between 0.1 and 0.8 is good and doesn't lead to over/undersegmentation. We decided to use minimal saliency equaling 0.25. The example of watershed segmentation is given on Fig. 11.

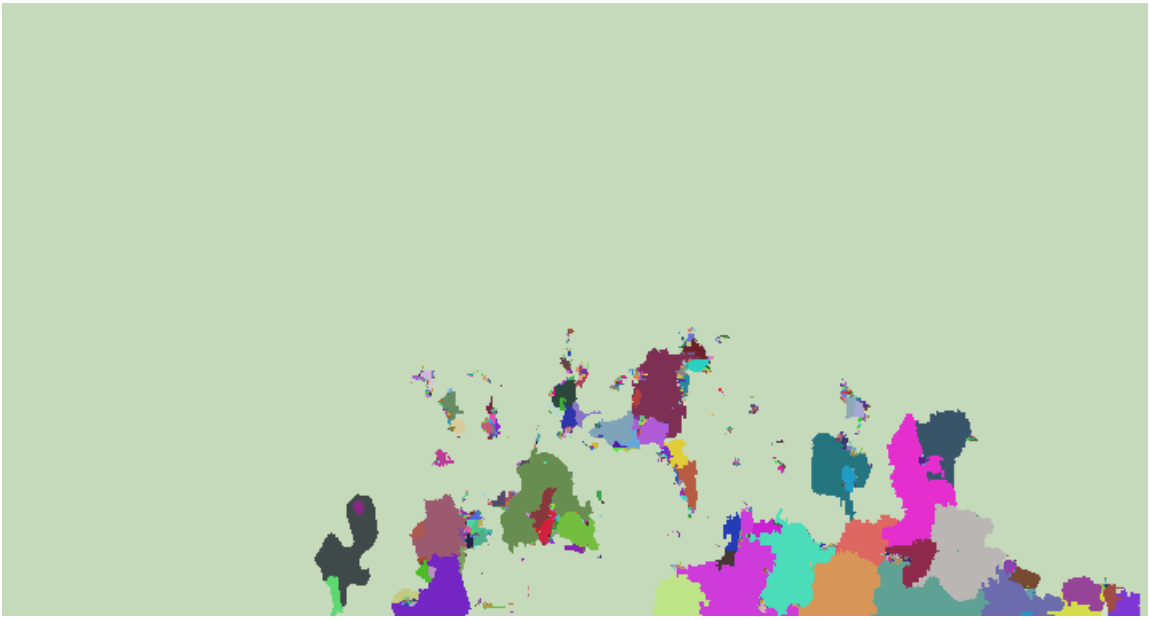


Fig. 11 Clustering of a flow image with extracted background using watershed method

5) Due to having tiny clusters, it is crucial to decide a threshold for a cluster size to be taken into consideration. We considered a pixel count per person as such a measure. To estimate the total count of people on the image, we used corresponding density map obtained from CSRNet. The total count equals to the sum of all its elements, and the pixel count per person equals to ratio of the product of image dimensions to total people count.
Then, clusters whose size is larger than the obtained threshold, are decided to be salient, and others are not.
6) For each salient cluster, the set of characteristics is calculated:
- its size (in pixels),
- averaged direction of its pixels,
- the distance between this averaged direction and the averaged direction within the whole image.

Averaged direction of a pixel set can be calculated as follows. According to the scheme on Fig. 6, we can map a vector to any hue by the formula:

$$\vec{v}(h) = (\cos 2\pi h, -\sin 2\pi h).$$

For a pixel set *S*, a set of vectors $\vec{v}_i = \vec{v}(h_i),\ h_i \in S,$ can be calculated. Then, they are all summed up:

$$\vec{v}_S = \sum_{h_i \in S} \vec{v}(h_i).$$

The $\vec{v}_S$ vector is converted back to a hue value then. It can be done by applying the arctangent function which takes into consideration the signs of its operands:

$$h' = \text{arctg}(x_S, y_S) = \text{arctg}\frac{y_S}{x_S},$$

where (x_S, y_S) are coordinates of $\vec{v}_S$, $h' \in [0, \pi]$ if $y_S \geq 0$, and $h' \in [-\pi, 0]$ if $y_S \leq 0$. Then, h' is converted to a hue value:

$$h = \begin{cases} -\dfrac{h'}{2\pi}, h' < 0, \\ \dfrac{h'}{2\pi} + \dfrac{1}{2} \text{ otherwise.} \end{cases}$$

7) If for a salient cluster there is a big enough distance between its averaged direction and averaged direction of the whole image, this cluster is declared as abnormal in terms of lined motion within it.

IV. Results And Discussion

For the experiment, we used HAJJ dataset [30]. It contains 18 videos, each having the duration of 20-25 seconds, with motion crowds. Both high dense and low dense crowds are represented in the dataset.

To each of the video, the above methodology has been applied. The maximal length of a non-major clusters' frames subsequence is 13, minimal one is 2, median one is 5.

For each video, a beginning frame from the longest sequence was considered. The overall results are given in Table 1. According to them, we decided to consider the deviation from averaged direction more than on 0.1 (which is equivalent to 36 degrees) to be abnormal. With such an assumption, we got 11 abnormal clusters total. On Fig. 12, the examples of abnormal clusters are given.

TABLE I
HAJJ DATASET FRAMES SUMMARY

Characteristic	Minimal value	Median Value	Maximal Value
Number of clusters	502	897	1164
Number of people	619.92	1019.73	1952.63
Pixels per person	351.12	1473.03	3369.71
Number of salient clusters	3	27	61
Maximal distance from averaged direction	0.071	0.106	0.232

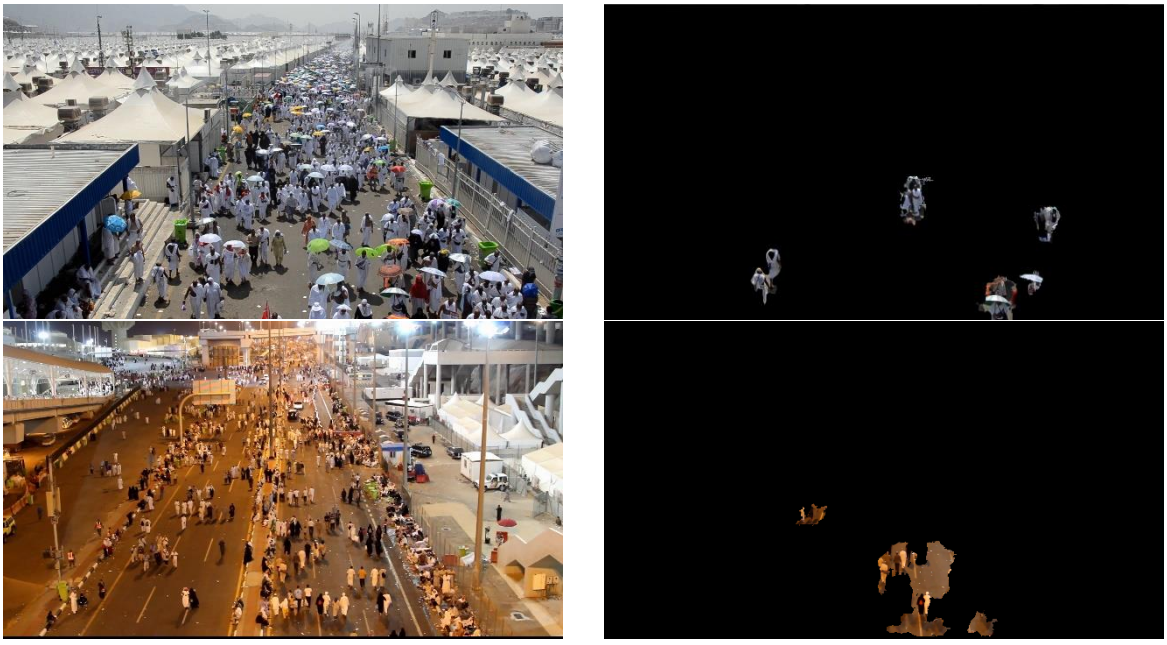


Fig. 12 Examples of abnormal movement direction detection on videos from the HAJJ dataset

REFERENCES

[1] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox, "FlowNet 2.0: Evolution of Optical Flow Estimation With Deep Networks," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2462-2470.

[2] Y. Li, X. Zhang, D. Chen, "CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1091-1100.

[3] C. Zhang, H. Li, X. Wang, X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 833-841, doi: 10.1109/CVPR.2015.7298684.

[4] J. Zhao, J. Li, Y. Cheng, L. Zhou, T. Sim, S. Yan, J. Feng, "Understanding Humans in Crowded Scenes: Deep Nested Adversarial Learning and A New Benchmark for Multi-Human Parsing," *ArXiv preprint*, 2018, arXiv:1804.03287.

[5] P. Zhang, W. Ouyang, P. Zhang, J. Xue, N. Zheng, "SR-LSTM: State Refinement for LSTM Towards Pedestrian Trajectory Prediction," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12085-12094.

[6] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint*, 2013, arXiv:1312.6203.

[7] T. N. Kipf, M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint*, 2016, arXiv:1609.02907.

[8] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, Y. N. Wu, "Multi-Agent Tensor Fusion for Contextual Trajectory Prediction," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12126-12134.
[9] Y. Liu, Q. Yan, A. Alahi, "Social NCE: Contrastive Learning of Socially-Aware Motion Representations," *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15118-15129.
[10] Y. Liu, R. Cadei, J. Schweizer, S. Bahmani, A. Alahi, "Towards Robust and Adaptive Motion Forecasting: A Causal Representation Perspective," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17081-17092.
[11] E. L. Andrade, S. Blunsden, R. B. Fisher, "Hidden Markov Models for Optical Flow Analysis in Crowds," *18th International Conference on Pattern Recognition (ICPR'06)*, 2006, Hong Kong, China, pp. 460-463, doi: 10.1109/ICPR.2006.621.
[12] E. L. Andrade, R. B. Fisher, "Simulation of crowd problems for computer vision," *First International Workshop on Crowd Simulation*, vol. 3, pp. 71–80, 2005.
[13] D. Ryan, S. Denman, C. Fookes, S. Sridharan, "Textures of optical flow for real-time anomaly detection in crowds," *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2011, Klagenfurt, Austria, pp. 230-235, doi: 10.1109/AVSS.2011.6027327.
[14] T. Singh, B.M. Singh, "Unusual Event Detection Using Energy of Motion Technique," *International Journal of Computer Science and Information Security (IJCSIS)*, Vol. 14, No. 11, November 2016, pp. 755-763.
[15] H. Chen, O. Nedzvedz, S. Ye, S. Ablameyko, "Crowd Abnormal Behaviour Identification Based on Integral Optical Flow in Video Surveillance Systems," *Informatica*, 2018, vol. 29, no. 2, pp. 211-232, doi: 10.15388/Informatica.2018.164.
[16] H. Chen, S. Ye, O. Nedzvedz, S. Ablameyko, "Application of Integral Optical Flow for Determining Crowd Movement from Video Images Obtained Using Video Surveillance Systems," Journal of Applied Spectroscopy, vol. 85, pp. 126–133, 2018. doi: 10.1007/s10812-018-0622-8.
[17] D. Helbing, P. Molnar, "Self-Organization Phenomena in Pedestrian Crowds", *ArXiv preprint*, 1998, arXiv:cond-mat/9806152.
[18] W. Yi, W. Wu, J. Li, X. Wang, X. Zheng, "An extended queueing model based on vision and morality for crowd evacuation," *Physica A: Statistical Mechanics and its Applications*, vol. 604, 127658, 2022.
[19] B. Solmaz, B. E. Moore, M. Shah, "Identifying Behaviors in Crowd Scenes Using Stability Analysis for Dynamical Systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 2064-2070, Oct. 2012, doi: 10.1109/TPAMI.2012.123.
[20] C. S. Soh, P. Raveendran, Z. Taha, "Automatic generation of self-organized virtual crowd using chaotic perturbation," *IEEE Region 10 Conference TENCON 2004*, Chiang Mai, Thailand, vol. 2, pp. 124-127, 2004, doi: 10.1109/TENCON.2004.1414547.
[21] P. Stein, A. Spalanzani, V. Santos, C. Laugier, "Leader following: A study on classification and selection," *Robotics and Autonomous Systems*, vol. 75, part A, 2016, pp. 79-95.
[22] M. A. Lopez-Carmona, A. P. Garcia, "Adaptive cell-based evacuation systems for leader-follower crowd evacuation," *Transportation Research. Part C: Emerging Technologies*, vol. 140, 103699, 2022.
[23] А. С. Акопов, Л. А. Бекларян, "Агентная модель поведения толпы при чрезвычайных ситуациях," *Автомат. и телемех.*, 2015, выпуск 10, с. 131–143 (*in Russ*.)
[24] S. Sholtanyuk, A. Leunikau, "Lightweight Deep Neural Networks for Dense Crowd Counting Estimation," *Pattern Recognition and Information Processing (PRIP'2021)*, Minsk, Belarus, 2021, pp. 61–64.
[25] Y. Zhang, D. Zhou, S. Chen, S. Gao, Y. Ma, "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 589-597.
[26] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbas, V. Golkov, P. v.d. Smagt, D. Cremers, T. Brox. "Flownet: Learning optical flow with convolutional networks," *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2758-2766.
[27] L. Vincent, P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 1991, no. 13, pp. 583–598.
[28] P. Jackway, "Gradient Watersheds in Morphological Scale-Space," *IEEE Transactions on Image Processing*, 1996, no. 5, pp. 913–921, doi: 10.1109/83.503908.
[29] S. V. Sholtanyuk, "Finding the optimal segmentation of a crowd image with watershed method," *International congress on Computer Science: Information Systems and Technologies (CSIST'2022)*, 2022, part 2, pp. 217-223.
[30] T. Alafif, B. Alzahrani, Y. Cao, R. Alotaibi, A. Barnawi, M. Chen, "Generative adversarial network based abnormal behavior detection in massive crowd videos: a Hajj case study," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 4077–4088, 2022, doi: 10.1007/s12652-021-03323-5.