

# Characterizing Anomalous Behavior of Moving Objects using Optical Flow Feature Extraction in Crowded Environment

Parul Saxena<sup>1</sup>, R S Jadon<sup>2</sup>

<sup>1</sup> Assistant Professor, MITS, Gwalior, MP, India

<sup>2</sup> Professor, MITS, Gwalior, MP, India

<sup>1</sup> gaurparul2007@gmail.com

**Abstract** - In this paper, a novel approach is proposed to detect an abnormal behaviour in a video. In order to provide security, it is necessary to analyze the behaviors of people and determine whether these behaviors are normal or abnormal. There are several keywords to refer to the abnormalities such as unusual, suspicious, anomalous and abnormal. In this paper, we refer opticalFlowFarneback method. After calculating optical flow and magnitudes using farneback method, we create an activity map using multiple frames to see the flow persistency over time. We have conducted experiments on various datasets and the results show that the proposed method provides 95% accuracy for UMN dataset.

**Keywords:** Abnormal crowd detection, Visual surveillance, Video analysis, Optical Flow, Real-time Applications

## I. INTRODUCTION

An anomaly is defined in Computer Science as a data value or a situation that does not follow the usually observed pattern and does not form a part of the data or pattern cluster. In the context of this research, an anomaly may be the out of order behavior of the crowd which can be characterized by activity of the scene. If localized activity is high then it can signify the clash between objects. These abnormalities can be observed over space or over time, respectively called spacial and temporal abnormalities [1]-[3]. If the abnormality happens in a certain area of the image where it is not expected to happen, it is called a spatial anomaly. For example, if a vehicle is moving in the pedestrian lane instead of the one meant for vehicles, it can cause a spatial anomaly.

When the change in the scene occurs over time, it is called a temporal anomaly. For example, if the vehicles suddenly start stopping in a fast lane after an accident, we can say that an anomalous behavior has been observed through time. In previous decades, the solution to these challenges has been published by many scholars, which addresses both temporal and spatial anomalies.

The behavior of a crowd using force based model was captured [4] in consideration of repulsive and attractive social forces between the crowd while in movement. They used models of pattern of motion encompassing both temporal and spatial and tried to design the behaviors with the use of Hidden Markov Models. Lagrangian Particle Trajectories [5] were used for modeling of flow of crowd and detecting invariants of chaos which capture the model behavior that is abnormal.

Studies like [6], [7] make use of dynamic texturing involving neural networks that are convolution, for modeling the behavior of crowds and calculations of denser trajectories and developing force models social in nature.

In this research, we have tried to implement a solution that is online, for detecting the anomalies in crowd behavior which are mainly caused by panic. Using the Farneback method, we first calculate the optical flow and the magnitudes and then we develop an activity map with the use of several frames to check the persistency of flow over time. Then this activity map is used to develop dual metrics of entropy and Temporal Occupancy Variation (TOV), respectively. These metrics are then employed along with a basic threshold for classifying every object into either normal or abnormal in the given sequence of frames. We discuss these in detail in the sections that follow.

## II. OPTICAL FLOW COMPUTATION USING FARNEBACK METHOD

Optical flow is the method of observing movement and speed of objects in a video frame or image. The Farneback algorithm is used to generate dense optical flow results based on a dense grid points. EstimateFlow helps farneback method to estimate the optical flow based on 2-frame motion that is the flow of the input frame  $i$  with respect to the preceding  $i-1$  frame. For the first run, this function sets the preceding frame to black frame and after that the new frame becomes the current frame.

The method generates an image pyramid indicates that every level has a low resolution as compared to the preceding level. With increase in the number of pyramid levels, we can handle more pixel motions among frames. It performs an iterative search to find the new location of the point. After finding the displacement, Gaussian filter is used to handle noise and helps in fast motion detection.

### III. CHARACTERIZING THE ANOMALIES USING LOW LEVEL FEATURES

As the number of surveillance cameras increases to cover both indoor-outdoor locations, the demand of intelligent system that detects anomalous events also increases. The aim of this intelligent system is to uncover events from a huge amount of videos to avoid dangerous conditions [8]-[11]. This is achieved by two video processing levels. First one is low level feature that helps in finding and describing the moving object within the scene. Though this step will not help in recognizing the action performed by the moving object and even not able to distinguish whether the behavior is normal or abnormal. So with the help of semantic information (second level), it is easy to identify the human action as well as the behaviour of a person (normal or not).

Behavior representation is the low level processing step of behavior analysis. It aims to capture related features describing the target object in the video. First, the interest region in the scene is detected based on low level features. Then, a description for this region (target object) is provided. Indeed, this level is complicated and challenging because it influences significantly the understanding of the interest object behaviour. In fact, the major challenge in behaviour representation is to find out appropriate features that are robust to several transformations for example changes in the background and on the object appearance. Global features are used to describe motion within the entire frame. Optical flow features are normally used to extract global motion information such as moving particles extraction to detect anomalous behavior in a crowded scene.

### IV. DESIGN AND IMPLEMENTATION

For implementing the study, we start by writing the code in MATLAB environment by use of toolbox for Computer Vision for calculating the flow of optical information, with ease. At first, we create an Object VideoReader, for reading the video as a stream of images. Then, we defined an object OpticalFlowFarneback, which inherits from the toolbox Computer Vision. This toolbox is then used to estimate the flow of optical object between neighbouring frames.

We base this estimation on the method of Farneback theory for Optical Flow. The paper written by Gunnar Farneback forms the basis for Farneback method. This gives us the algorithm for motion detection. To begin with the step one, this algorithm uses the transform for polynomial expansion for approximating every neighbourhood for both the frames using quadratic polynomials. After further refinements serially, Farneback estimates the fields of displacement from coefficients of polynomial expansion.

Thereafter, for every frame in the video, we convert each frame into a grayscale image before applying a Gaussian filter for smoothening of the image. In the next step, we calculate the flow of the optics by using the method for estimateFlow(opticalFlowFarneback, frameGray), which is a part of the Vision Toolbox of the Computer. This process is similar to the method including CV2.CalcOpticalFlowFarneback(), which is a part of OpenCV.

The flow object, as a result has two main features: Magnitude and Angle. The property that is more important for the purpose of this research is the magnitude property. This is because we are more conscious of the magnitude of change in the optical flow between consecutive objects for detection of a situation of alarm. If the magnitude is high, it means that it is more probable for us to detect an anomaly or an alarming situation. If the magnitude is less, we can infer that the crowd is in fact walking at a slower pace.

Once we get the magnitude section of the object flow, we set the threshold of the magnitude at  $> 0.3$  for getting rid of the noise. This part of the process is more important as there may be few areas in the object where the change of flow may be estimated but there may not be any abnormal behavior in the crowd and hence no anomaly. In the next step, we create a three dimensional matrix where the z-axis is of length 30 and it represents every frame's image of magnitude. The use of matrix with 30 frames in the z-axis is due to the previous research in the field, which concludes that the capture of the dataset is through cameras that are enabled for capturing 30 fps or frames per second.

Due to this, we can hold a second from the video or the sequence into a three dimensional matrix that we are using. After acquiring the matrix of magnitude, we can achieve the frames' average to obtain the activity map. This activity map shall have a color ranging from black to white, where white color will mean that there is a movement in that particular portion of the image that we consider as objects. On the other hand, black will mean that there is a lack of movement in that area and there is no change in pixels.

Gray pixels show that the movement in those areas is slower. For exonerating of the pepper and salt noises in the map of activities, we can apply the process of median filtering on the map. This gives us the sequence of activity maps, wherein now we are able to detect the excessive occurrence of motion in them.

To achieve this step, we reduce every currently occurring activity map from the previous activity map for getting a matrix of difference. In previous studies about this topic, this matrix is termed as TOV or Temporal Occupancy Variation. The frame for TOV simply depicts the difference in pixels between two consecutively occurring activity maps. This means that now we can see the motion as it happens. In my code, this is called the dif. The current research suggests dual approached to detect the alarm situation at this point.

In case one of the approaches shows the presence of an alarm situation, then whatever the situation of the other parameter, that is, even if it shows an alarm or doesn't show one, we can still detect the abnormality. For this, the first of the two ways is to count the presence of white pixels in the frame of difference. This shows us the intensity or the magnitude of the difference. This difference is called the TOV difference.

On the other hand, entropy can also be helpful for us in estimating the size of the motion in the frame of difference. Hence, we make use of entropy or the dif function in MATLAB for obtaining the information of entropy of the object. In my code, `ImgEntropy` is the variable for this. The formula below shows the process of calculation of the entropy for an image, where the pixel frequency with a specified intensity value is known as  $P_i$  and it ranges from 0 to 255.

$$H = - \sum_i p_i (\log_2 p_i)$$

In the activity map thus obtained, if there is an increase in the motion, the entropy increases simultaneously, generally all the points will be black in case of an absence of motion. In conclusion, we apply the smoothing exponentially on both `ImgEntropy` and `TOVDifference` for obtaining results that are more reliable and which are sustained by the previous values of these tangibles. With this step, we can prevent our code from detecting any alarm in the frames when there is no alarm situation.

In my code, the exponential smoothing outputs are known as `E_Level` and `TOV_Level`. In the last step, we perform a check to see if these exceed a particular threshold or it remains within the threshold. If either of these get higher than the threshold, then it may infer a situation of alarm. This may happen when a crowd of object starts to run suddenly. For each dataset, there are unique regions of threshold and therefore the optimal thresholds are unprecedented and these do not hold true for other datasets as the environmental conditions like illumination and the distance of the cameras may be unique for each dataset.

As our code may work in a real time online environment, we can insert an 'ALARM' string immediately into the video playing. After each video ends, we can see immediate changed on the `E_Level` and `TOV_Level`. In the next section, we discuss these graphs in detail. From our pipeline, we present a few snapshots for the reference.

For the dataset for input, we have used a dataset of UMN [12]. This dataset takes into account three scenes which are different, namely Indoor, Lawn and Plaza. It includes various videos from each type of the scene. Given below are the scene pipeline snapshots for every scene. The graphs resulting from these images are discussed in the next section.

#### Lawn Sequence of UMN Dataset

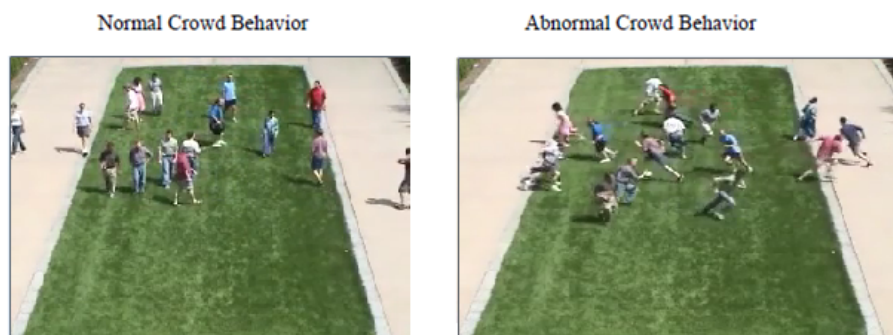


Figure 1. RGB Image of Lawn

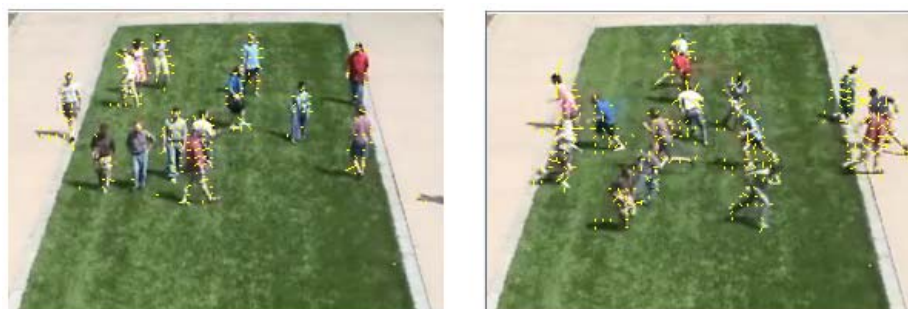


Figure 2. Optical Flow Output of Farneback

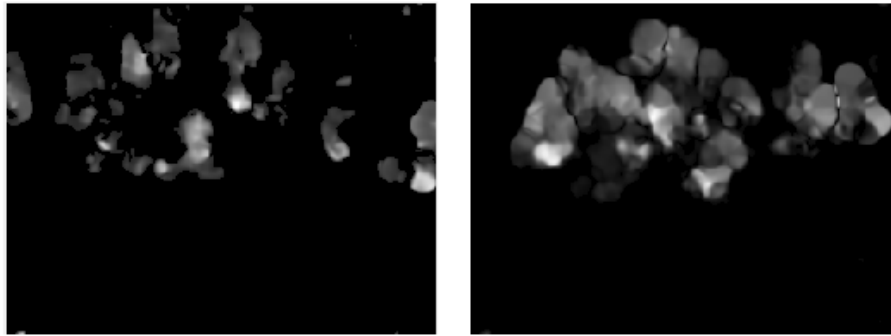


Figure 3. Optical Flow Magnitude

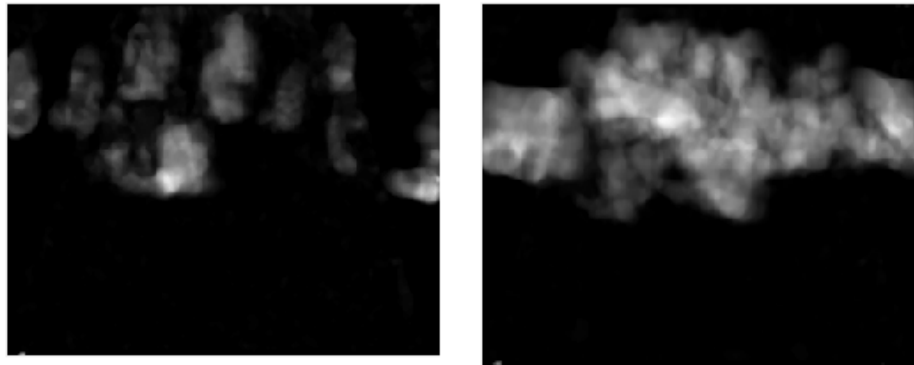


Figure 4. Maps of Activity



Figure 5. Difference between Simultaneous Activity Maps



Figure 6. Code Output Showing Anomaly Detection

### Indoor Sequence of UMN Dataset



Figure 7. Indoor Image in RGB



Figure 8. Optical Flow Output of Farneback

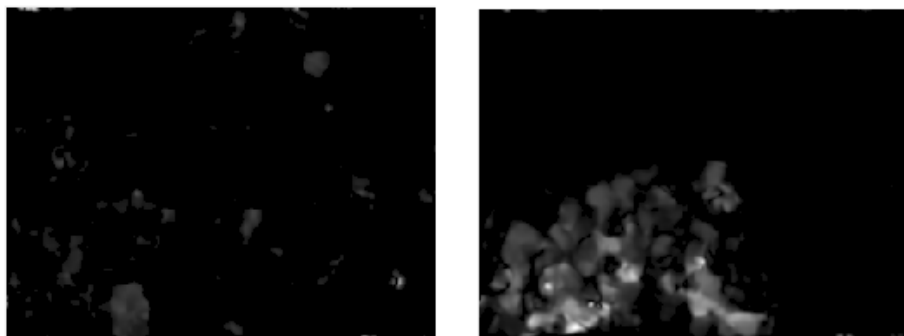


Figure 9. Optical Flow Magnitude

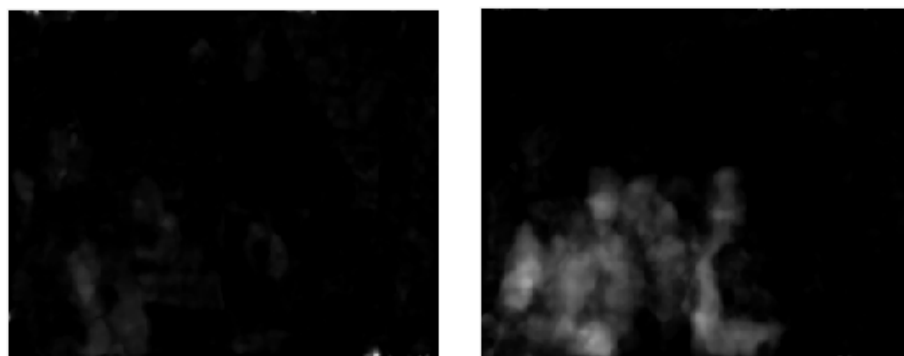


Figure 10. Maps of Activity



Figure 11. Difference between Simultaneous Activity Maps



Figure 12. Code Output Depicting Alarm Detection

Plaza Sequence of UMN Dataset

Normal Crowd Behavior



Abnormal Crowd Behavior



Figure 13. Plaza Image in RGB



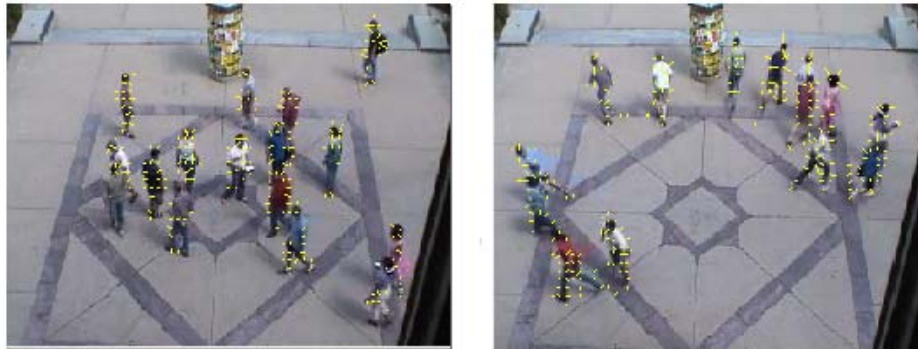


Figure 14. Optical Flow Output of Farneback

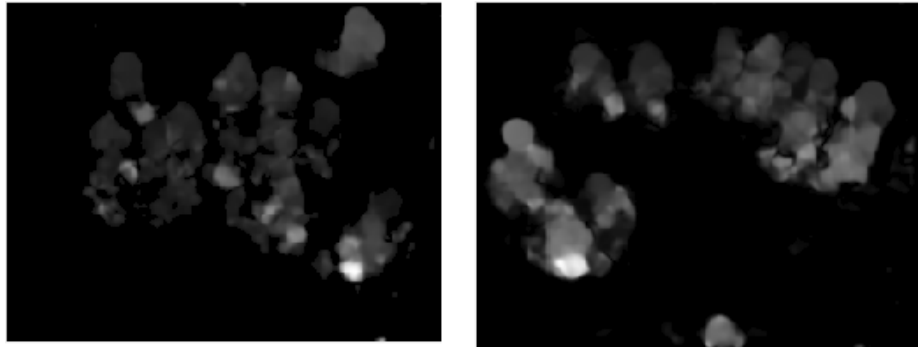


Figure 15. Optical Flow Magnitude

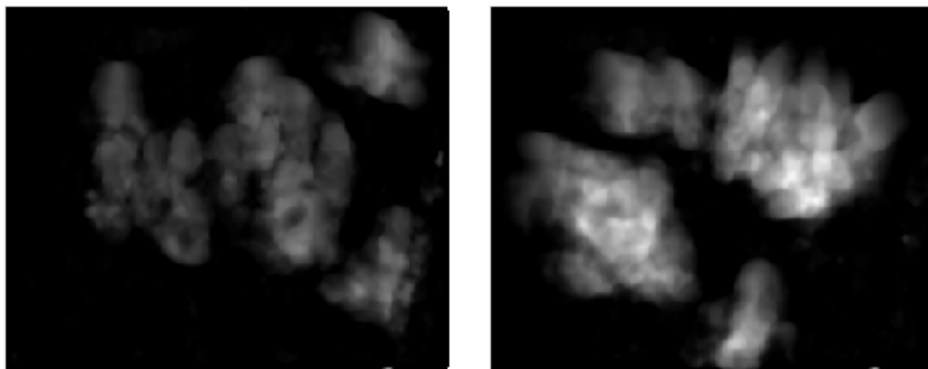


Figure 16. Maps of Activity

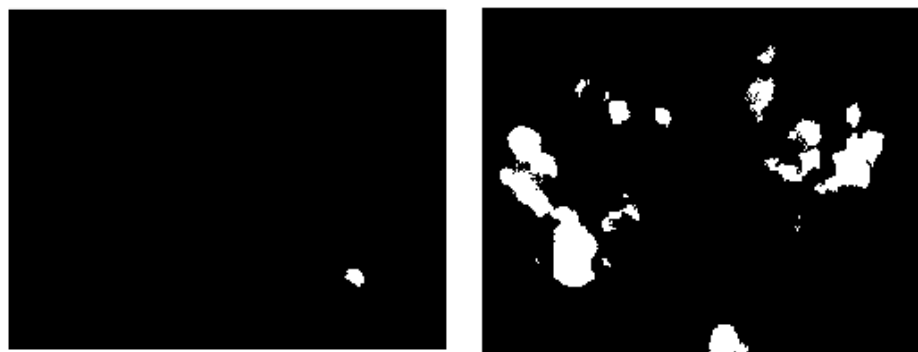


Figure 17. Difference between Simultaneous Activity Maps



Figure 18. Code Output Showing Anomaly Detection

*Broad Analysis of the above attached code's results:* In the first row of every sequence, as in figure 13 we can see the image frames that are raw. The second row of every sequence, as shown in figure 14, we see the optical flow in between the frames in the row one and the preceding row to these. As is seen in these sequences, the angle and magnitude of the flow of optics is shown by the arrows in yellow. If the amount of motion is higher, the yellow arrow is longer. Therefore, in all the scenes of alarm, the magnitude of yellow arrows is bigger and in scenes where the actions are normal, the magnitude of yellow arrows is smaller.

In the third sequence of every output, as in figure 15, we show only the flow's magnitude, which is the beginning point of accessing a situation of alarm, as described in the description of code. In the fourth row of every sequence, as in figure 16, we can see the activity maps of every 30 frame sequence. As we can see, there is a large part occupied by the white region in the one that is abnormal in comparison to the one that is normal, as there is more difference in the events that are abnormal.

While getting the differences between the respective activity maps, we can observe an abnormal event alarm situation exactly in the frames where the crowd stops or has a slower pace or starts running. This creates a large area of a frame with white pixels as the crowd with a normal pace does not have many whites and the one that is abnormal, has a large number of white pixels. In this case, it is sufficient for us to assume an alarm situation, after a certain threshold is reached. In figure 18, we can see the output of the code.

## V. EXPERIMENTAL RESULTS

We see some samples of Entropy Metrics and TOV below, which are calculated one time for every type of scene, like Lawn or Indoor or Plaza. The x-axis denoted the number of the frame and the y-axis is the TOV between two activity maps that are consecutive and the entropy of the two activity maps' difference, respectively. As the data for ground truth is unavailable, we can observe this on every video and we use this in our calculations. The occurrence of an alarming situation is indicated by the blue lines, like when the crowd attempts to escape. Until the video comes to an end for every recording, the situation of alarm lasts till the end. In the images that follow, we can find that each of the two metrics yields the outputs that are very similar as their calculations and meanings are a lot similar as well. Both of them indicate the flow changes that happen over time.

We can notice easily the start of the anomaly and thereafter the rapid increase in the metrics, which create a pulse which lasts till the end of the situation. This depicts the promising nature of both the metrics under consideration, for classifying the object into abnormal and normal categories.

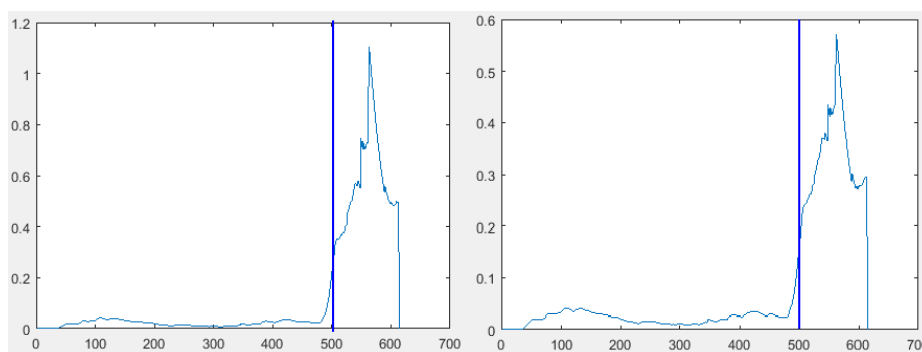


Figure 19. Lawn Scene TOV and Entropy Plots, respectively



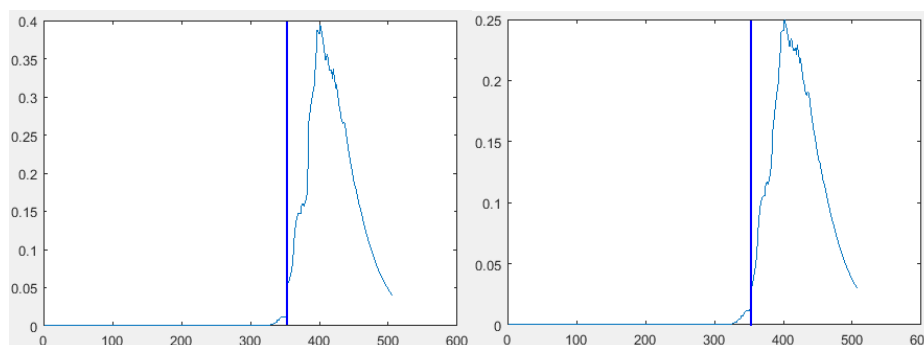


Figure 20. Indoor Scene TOV and Entropy Plots, respectively

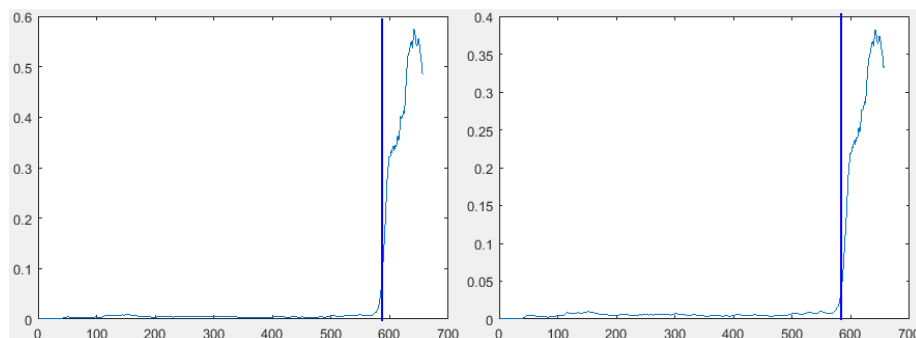


Figure 21. Plaza Scene TOV and Entropy Plots, respectively

It is very easy to see the point of occurrence of anomalies with the human eye in accordance with the metrics in consideration. Even then, the important point here is the choice of the threshold values for classification of every object frame as either a normal or an abnormal one. At first, as the angles, elevations and distances of the cameras used for surveillance in every scene is different from another scene, the regions that are feasible for the thresholds may not be similar. From this, we can infer that the best values of threshold will be different for every scene group. For overcoming these issues, the extreme values that are maximum or minimum for each of the two metrics are learnt as a function of the elevation of the angle of viewing and afterwards this learner is able to be introduced automatically into new cameras for surveillances that are in the process of installation.

Even when the parameters here are calculated with techniques of cross-validation, it may not be sufficient for evaluating accurately. A more accurate indicator of performance for similar classifiers is the ROC (Receiver Operating Characteristics) curves. These ROC curves depict the relationship between specificity and sensitivity, which are accurate measures of performance overall. Therefore we focus our efforts on three separate intervals of threshold for every type of scene that needs to be plotted for the ROC curves.

The Entropy metrics and TOV thresholds are simultaneously changed, while consideration of the relative scale among them. Given below are the plots for ROC for every type of scene for an average of similar type of sequences of video:

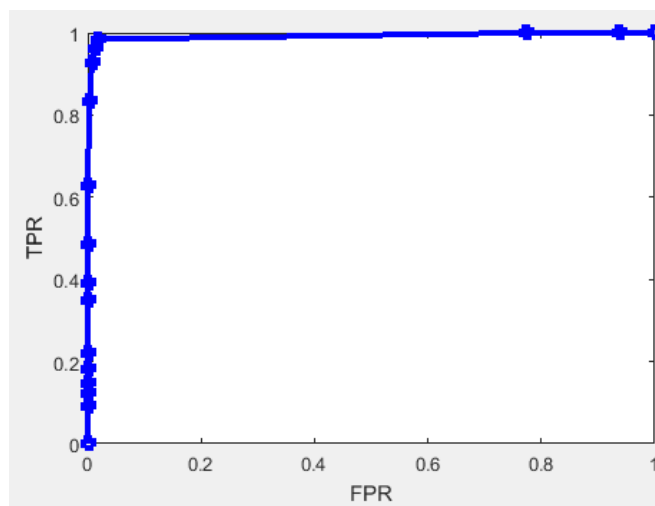


Figure 22. Lawn Scene ROC Plot

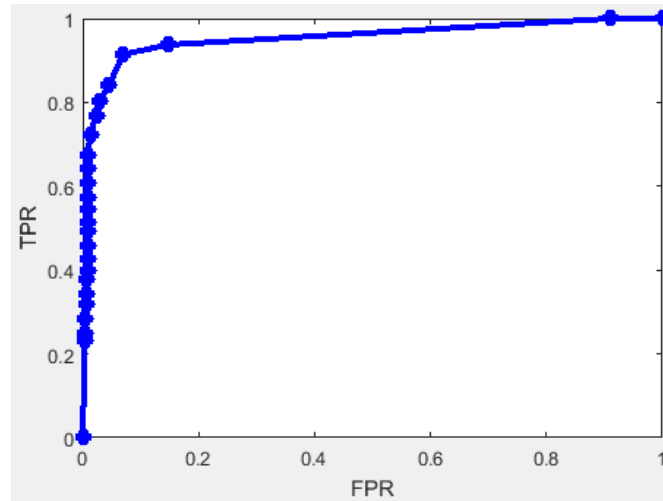


Figure 23. Indoor Scene ROC plot

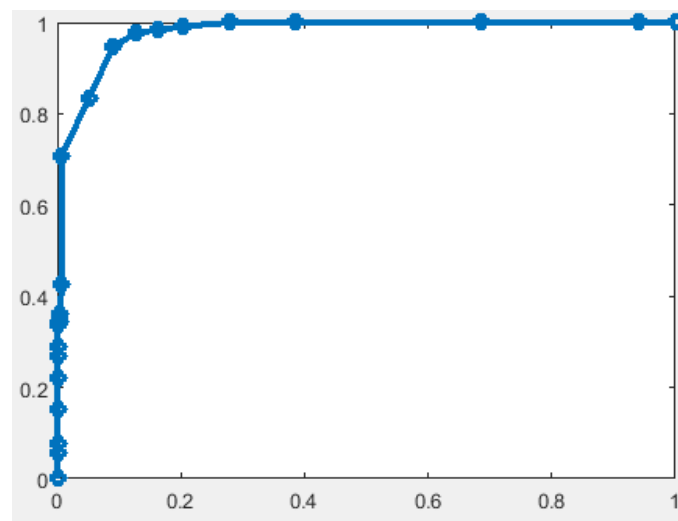


Figure 24. Plaza Scene ROC plot

The AUC (Area Under Curve) is an indicator of the performance for plots for ROC, as we can see from the figures. All of these performances lay above the 95 percent threshold, which is a positive factor for detecting anomalies. Various differing values of threshold give us differing specificities and sensitivities. In case it is more mandatory to detect every event that is abnormal, then the TPR sensitivity must be very high too. In case it is more important to avoid detection of more false alarms, then we must set a higher value of threshold relatively. In this case, we need to increase the specificity, which is equal to 1-FPR.

## VI. SUMMARY AND CONCLUSION

In our pipeline, the most important property is that it has to work in real time as it has to be implemented on cameras for surveillance, where real time visuals are captured. Afterwards, the rate of processing of the algorithm must be faster than the sequence of input in itself. The algorithm implemented by us has a performance with a rate of processing above 30 frames per second and when it is online, it still can take the input at 30 frames per second too.

For the human eye, 30 fps is a very satisfying visual. We can make this performance possible due to two features considered for every frame, like Entropy or TOV, which can have a real time updating. This algorithm works on the basis of the magnitudes of the optical flow. From this, we can deduce that wherever there is a large amount of motion to be estimated between two frames that are consecutive, there must appear an alarm situation. Our algorithm's feature space is based on the Entropy and TOV, thus allowing this code to run in real time online environment. The input frames must be processed faster by us as we need the calculation to be online and in real time.

To obtain features elaborately, the cameras being used should have a high resolution and as the cameras for surveillance are generally with a lower resolution, it is not possible for us to easily inspect peoples' behavior from the frames obtained from these low resolution surveillance cameras. Another issue that is important is the decision for threshold value for various cameras located at different places. These threshold values are the values that we rely on completely for detection of alarm situations. To get the best results for specific cameras, we must pick them carefully. Nonetheless, every camera is placed at various different elevations, having varying angles of viewing.

Our aim in this work has been to calculate the metrics for detecting anomalies and abnormalities in a given sequence of images, obtained from surveillance cameras. This has a high importance, as day by day, the concerns for safety have been increasing in today's world. This algorithm strives to achieve the automation of the observation process for events in case of the surveillance cameras, in as accurate manner as possible.

The input to this system are the records of various surveillance cameras, which is what this method strives to achieve. After studying the source paper, we calculated the optical flow and generated the activity map for enabling and capturing the changes in the optical flow in a given time frame. This activity map is then used to calculate the entropy or the factor of surprise in the situation and then the variation of temporal occupancy is calculated. By using the entropy changes and variation in temporal occupancy, each frame of video is then classified into normal or abnormal categories. The ROC plots depict that the performance of the classification is as expected, and hence successful.

The efficiency and validity of the framework proposed is verified by the use of popular datasets that compare this system's performance with similar methods which work on the same dataset.

## REFERENCES

- [1] L. Kratz and, K. Nishino (2009). Anomaly Detection in Extremely Crowded Scenes Using Spatiotemporal Motion Pattern Models. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1446–1453.
- [2] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos (June 2010). Anomaly Detection in Crowded Scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1975 - 1981.
- [3] J. Kim, and K. Grauman (2009). Observe Locally, Infer Globally: A Space-Time MRF for Detecting Abnormal Activities with Incremental Updates. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2928.
- [4] R. Mehran, A. Oyama, and M. Shah (2009). Abnormal Crowd Behavior Detection Using Social Force Model. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.
- [5] S. Wu, B. Moore, and M. Shah (2010). Chaotic Invariants of Lagrangian Particle Trajectories for Anomaly Detection in Crowded Scenes. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.
- [6] W. Li, V. Mahadevan, and N. Vasconcelos (2014). Anomaly detection and localization in crowded scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 36(1),18–32.
- [7] Shifu Zhou, Wei Shen, Dan Zeng, Mei Fang (2016). Spatial-temporal Convolutional Neural Networks for Anomaly Detection and Localization in Crowded Scenes. *Signal Processing Image Communication*.
- [8] W. Hu, T. Tan, L. Wang, S. Maybank (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 34 (3) 334–352.
- [9] Ali Khaleghi, Mohammad Shahram Moin (2018). Improved Anomaly Detection in Surveillance Videos Based on A Deep Learning Method. *Artificial Intelligence and Robotics (IRANOPEN) in IEEE*, 73-81.
- [10] Dan Xu, Yan Yan, Elisa Ricci, Nicu Sebe (26 oct. 2016). Detecting Anomalous Events in Videos by Learning Deep Representations of Appearance and Motion. *Computer Vision and Image Understanding*.
- [11] O. P. Popoola, K. Wang (2012). Video-based abnormal human behavior recognition - a review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 42 (6) 865–878.
- [12] "Univ. Minnesota. unusual crowd activity dataset of university of minnesota." [Online]. Available: <http://mha.cs.umn.edu/movies/crowdactivity-all.avi>