A New Pattern Recognition Method for Abnormal Event Detection in Crowded Scenarios

Supervisor: Dr. Md. Haider Ali

Co-Supervisor: Dr. Jia Uddin

Tahjid Ashfaque Mostafa 13101098

Department of Computer Science and Engineering

BRAC University

Submitted on: 18th April 2017

DECLARATION

We, hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researchers are mentioned by reference. This Thesis, neither in whole or in part, has been previously submitted for any degree.

Signature of Supervisor

Dr. Md. Haider Ali

Signature of Author

Tahjid Ashfaque Mostafa

Signature of Co-Supervisor

Dr. Jia Uddin

ACKNOWLEDGEMENTS

I am grateful for the influences and support given to me by my professors, friends and family members, who helped, motivated and encouraged me in every step my work.

First of all I would like to express my heartfelt gratitude to my advisors Dr. Md. Haider Ali and Dr. Jia Uddin for their continuous support, encouragement and guidance throughout my research.

They competently directed my work and provided significant support on countless occasions, instilling their enthusiasm in me. This work would not be completed without their supervision, guidance uncountable ideas. It has been a huge honor for me to have gotten a chance to work with them at BRAC University.

I would like to thank all the faculty and staff members of BRAC University who, throughout the years, have provided an incredible learning environment and helped me grow as a person.

Finally I would like to thank all my friends and family members who were always there for me and supported and encouraged me along the way.

.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

We propose an autonomous video surveillance system which analyzes surveillance footages of extremely crowded scenes and detects abnormal events. For any particular scenario, any event that diverts from the usual pattern can be classified as an abnormal event. The model analyzes the local spatial-temporal motion pattern and detects abnormal motion variations and sudden changes. It can be divided into two major parts, selecting a set of Points of Interest (POI) from given frames using ORB (Oriented FAST and Rotated BRIEF) feature detector and tracking them across multiple frames and dividing the input video frame in a number of cubes and track the motion patterns in each of the cubes for spatial-temporal statistical deviations. To evaluate the performance of proposed model we utilize several datasets and compare the acquired results of the proposed model with various state-of-the art models. Experimental results demonstrate that the proposed model outperforms the other models by exhibiting an average of 96.12% accuracy using Convolutional Neural Network.

CHAPTER 01

INTRODUCTION

1.1 Motivations

Public safety has become a major concern in modern times. Public areas such as airports, parks, hospitals, shopping malls are often closely monitored for any signs of unusual activities. With the recent decrease in the cost of video surveillance equipment, a large number of areas can be monitored constantly. However monitoring a place for abnormal and emergency situations is quite meaningless unless the situation can be identified and appropriate responses can be taken. Extremely crowded scenes require monitoring a large number of individuals engaged in various arbitrary actions, which is a significant challenge even for a human observer. It might include hundreds of people scattered across the frame and possibly thousands of individuals in the whole video sequence with extremely irregular motion patterns which might cause occlusions. The computation approach must be able to detect abnormal events in any specific area of the frame while maintaining the structural view of the entire scene. And the large number of subjects' cause analyzing the actions of the individual subjects to be a massive challenge, even with the computational prowess of the modern computers.

Abnormal events can be both spatial, an event which is abnormal in context of its surroundings and temporal, which is an event over duration of time which is abnormal [1]. It can also be divided into Local Abnormal Events where the behavior of an individual is different from its neighbors and Global Abnormal Events where the whole scenario is abnormal [2]. It takes sufficient

computing power in order to analyze and detect the nature of an event in real time, because if the event cannot be identified in real time, the purpose of surveillance is not served.

In this paper, we proposed a model for identifying abnormal events in extremely crowded scenarios by analyzing the motion patterns of a set of defined features and combining that with the overall motion structure within the frame. The symbols used throughout the paper are listed in Table 1.

Table 1 – Symbol Table

Symbol	Definition
U	Original color image
p, q	Pixel Coordinates
$NLu(p)$	Non Local Means Denoising of image at point p.
$d(B(p),B(q))$	Euclidean distance between the image patches with centers respectively at p and q
$C(p)$	Normalizing factor
F	Decreasing function for $NLu(p)$
$B(p, r)$	a region with center at p and size $(2r + 1)^2$ pixels
$w(p, q)$	Weights computed using an exponential kernel
σ	The standard deviation of the noise
H	The filtering parameter set depending on the value of σ
M	Moments of the patch for determining corner properties.
$I(j, k)$	Intensity of the image at point j and k
C	Centroid
$\overrightarrow{OC}$	Vector from corner O to the centroid.
$atan2$	Quadrant aware version of **arctan.**
G	Radius of the circular region holding j *and* k.
N	Maximum number of detected features
τ	Binary test for smoothed u.
$u(j)$	The intensity of u at point j
T	Number of binary tests used.

V	Set of detected features for all frames.
x_i *and* y_i	Co-ordinates for feature i
d_i	The distance between the feature in the previous frame the matched feature in current frame
θ_i	The orientation or angle of the feature
r_i	The response by which the strongest key points has been selected
R_p	Spatial-Temporal Gradient determined using the Sobel derivative function for every pixel p in non-overlapping region R for k frames.
V	Horizontal dimension of the video.
W	Vertical dimension of the video.
Z	Temporal dimension of the video.
N	Total number of pixels.
$G(\alpha, \beta)$	Three dimensional Gaussian Distribution.
S	Final feature vector

1.2 Contribution Summary

The notable contributions can be summarized as follows:

- Increase detection accuracy by removing noise with Non-Local Means (NLM) denoising
- Separating background scenarios and foreground objects for better object tracking.
- Using motion heatmap as a mask for feature detection to better understand motion patterns.
- Use of open source ORB feature detector which leads to increased accuracy in case of feature identification and tracking since ORB is rotation invariant and more efficient compared to other feature detectors like SIFT or SURF [22].
- Using 3D Gaussian distribution to better portray the underlying motion structure of the whole video.

1.3 Thesis Orientation

The rest of the thesis is organized as follows:

- Chapter 02 includes the necessary background information regarding the proposed approach.
- Chapter 03 presents the proposed model of our approach.
- Chapter 04 demonstrates the experimental results and comparison.
- Chapter 05 concludes the thesis and states the future research directions.

CHAPTER 02

RELATED WORKS

Video surveillance has been a major topic for research in recent years. Many approaches have been undertaken in analyzing and classifying video events. For example, object tracking [3], pedestrian detection [4], crowd counting [5], background modeling [6], action detection [7] etc. Estimation of crowd density based on texture and motion area ratio has been proposed as well by various researchers [8, 9, 10]. Similarly, abnormal event detection has also attracted major attention in recent years. Abnormal event detection can mainly be divided in three types [1], which are

- Macroscopic Models: Models that density and velocity of the whole crowd,
- Microscopic Models: Models that focus on individual behavior,
- Hybrid Model: Models which consider both overall and individual behavior.

Macroscopic models are used to detect Global Abnormal Events (GAE) as they deal with the density and velocity of the whole frame and Microscopic models can detect Local Abnormal Events (LAE) since they focus on behavior of the individual subjects within the frame. The usual approach is to first define the normal behavior and its properties, and then classifying those behaviors which do not have similar properties as the predefined normal behavior as abnormal events, because not many examples of abnormal behavior are available, and also it is impossible to completely determine the types of abnormal events that might occur. So it makes more sense to determine the properties of normal events, which occur frequently. Once the properties and patterns of the usual events is identified, it then becomes easier to classify events which defer from the usual patterns to be abnormal.

Many methods analyze the trajectory of the individuals to classify event types with many well developed methods such as HOG [5] or motion patterns and appearances [11]. Mehran *et al.* used an optical flow based social force model and then used Latent Dirichlet Allocation (LDA) to detect abnormality [12]. Hua *et al* [13] used Normalization Cut Clustering and Benezeth *et al.*used a 3D spatio-temporal foreground mask fusing Markov Random Field [14]. Nacim *et al.* [15] used a heatmap image as a mask for Harris Corner Detector to identify key Points of Interest. Lagrangian Particle Dynamics is used in [16] for detection of flow instabilities which is very helpful in segmentation of high density crowd flows. A combination of Hidden Markov Models (HMM), spectral clustering and principal component analysis is used in [17, 18] to detect emergency scenarios.

CHAPTER 03

PROPOSED FEATURE EXTRACTION MODEL

3.1 Introduction

The proposed model can be divided into four major parts, namely:

- Input preprocessing- which includes :
    - Extracting frames from the input video
    - Denoising the frames using NLM denoising
    - Separating background scenes and foreground objects
    - Creating a motion heatmap to detect motion activity within the frame.
- Feature detection and tracking them across multiple frames.
- Dividing the frame into uniform spatial-temporal cubes and detecting the underlying motion pattern.
- Generating a set of feature vectors and defining a criteria to separate normal and abnormal events.

Figure 1 shows the steps of our proposed model in a block diagram form.

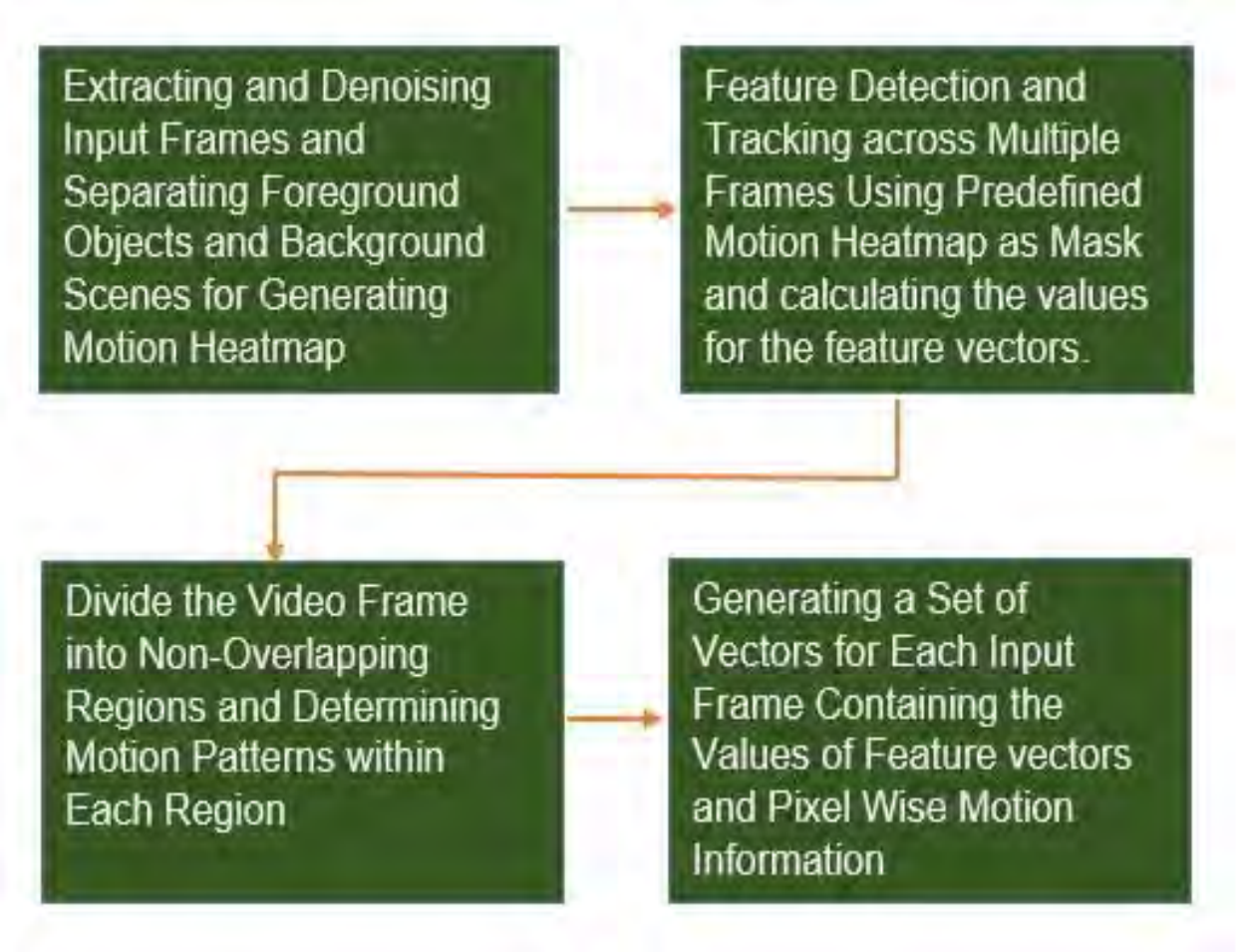


Figure 1: Block Diagram depicting the steps of our proposed model

3.2 Input Preprocessing

The surveillance video used as input is put through some preprocessing steps before it can be used for feature detection and tracking. These steps help increase the accuracy of the result and decrease the computation time. The steps are mainly as follows:

3.2.1 Input Frame Extraction and Denoising with Non Local Means

Surveillance videos will possibly contain a lot of static and dynamic signal distortion and blur, commonly known as noise, which might affect the accuracy of analysis. Noise represent a set of random pixels which do not represent the features of the scene being captured. Usually, for any given camera, low light conditions will likely result in greater amount of noise. Surveillance videos most often do not have the ideal shooting conditions, they contain footages from day, night, rainy or showy conditions. We use pixel wise Non Local Means (NLM) denoising to decrease the amount of blur and noise in the frames [19]. The mathematical equation of NLM is:

$$NLu(p) = \frac{1}{C(p)} \int f(d(B(p), B(q))u(q)\, dq \tag{1}$$

Where, ***d(B(p),B(q))*** represents the Euclidean distance between the image patches with centers respectively at ***p*** and ***q, C(p)*** is the Normalizing factor and ***f*** represents the decreasing function. The denoising of a color image ***u = (u1, u2, u3)*** and pixel ***p*** gives us:

$$\hat{u}_i(p) = \frac{1}{C(p)} \sum_{q \epsilon B(p,r)} u_i(q) w(p,q) \quad and \quad C(p) = \sum_{q \epsilon B(p,r)} w(p,q) \tag{2}$$

Where ***i*** = 1, 2, 3 and ***B(p, r)*** represents a region with center at ***p*** and size ***(2r + 1)²***pixels.

w (p, q) depends on the squared Euclidean distance ***d² = d²(B (p, f), B (q, f))*** of the ***(2f + 1)²*** color patches centered at ***p*** and ***q.***

$$d^2\big(B(p,f), B(q,f)\big) = \frac{1}{3(2f+1)^2} \sum_{i=1}^{3} \sum_{j \epsilon B(0,f)} \big(u_i(p+j) - u_i(q+j)\big)^2 \tag{3}$$

Each of the pixels is replaced with an average value of the most resembling pixels. Weights ***w(p, q)*** is computed using an exponential kernel.

$$w(p,q) = e^{-\frac{\max(d^2-2\sigma^2,0.0)}{h^2}} \quad (4)$$

Where σ represents the standard deviation of the noise and ***h*** is the filtering parameter set depending on the value of σ. Figure 2 gives an example of how NLM denoising works

(a) **(b)**

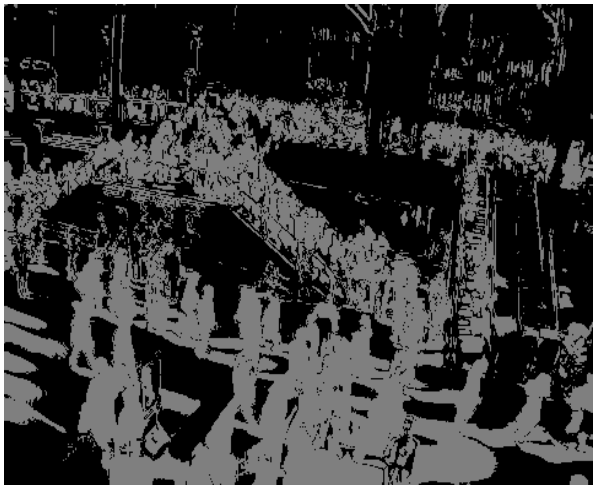
(c) **(d)**

Figure 2: (a) A sample scenario, (b) Figure 2(a) after denoising by NLM, (c) Another sample scene, (d) Figure 2(c) after denoising by NLM

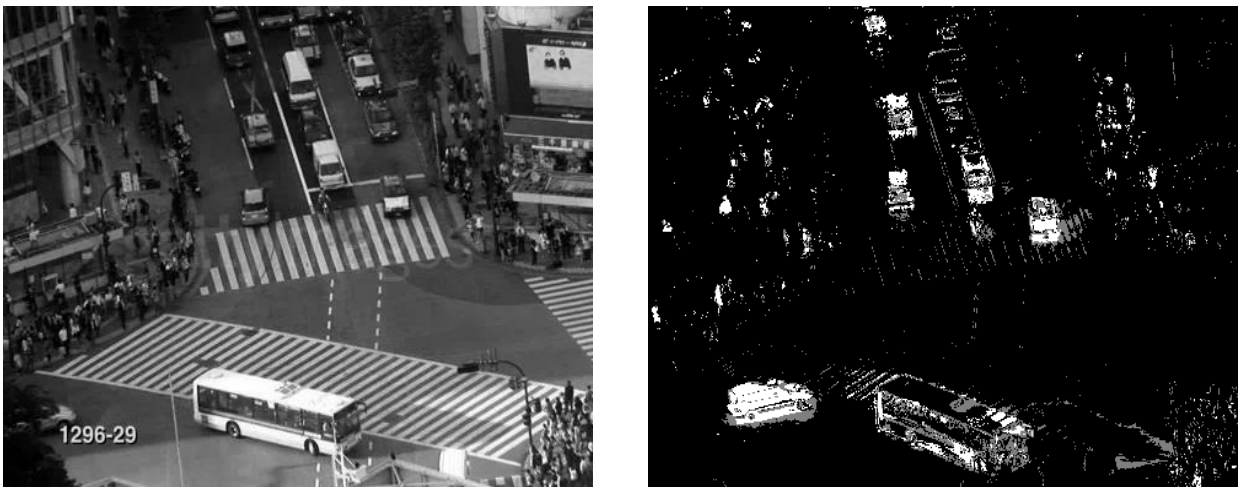
3.2.2 Application of Mixture of Gaussians Method for Background and Foreground Separation

Separating the background from the foreground is a pivotal part of any event detection procedure. The abnormal event most often concerns subjects in the foreground such as people or vehicles, so it is imperative to correctly separate foreground and background. In this model we use a Gaussian Mixture-based Background [20] and Foreground segmentation algorithm [21]. This model selects appropriate number of Gaussian distributions for each pixel, which provides better adaptability for varying scenes due to illumination and other changes. This also detects shadows and separates them thus enabling us to ignore the movement of shadows in construction of motion heatmap and improving accuracy. Figure 3 shows an example of how our background/foreground segmentation algorithm works.

(a)



(b)



(c) **(d)**

Figure 3: (a) A sample scenario, (b) After background subtraction of Figure 3(a), (c) Another sample scenario, (d) After background subtraction of Figure 3(b)

3.2.3 Motion Heatmap Generation for Mapping Motion Activity

Motion heatmap refers to a 2D histogram that indicates and highlights the regions within the frame that experienced any kind of motion activity in the video. Binary blobs of moving objects are extracted after background and foreground separation and later accumulated to form the motion heatmap. Region of Interest (ROI) for the next step is defined from the obtained heatmap which is used as a mask to increase the quality of the result and reduce computation time, especially in case of videos with long duration. The definition of what is normal and what is not might vary depending on the context, i.e. day/night, peak/off-peak hours etc. For example, low traffic on a road during night is considered to be normal, but during office hours it would be considered abnormal. A motion heatmap needed to be built for all the available sample scenarios to improve the performance of the model. Figure 4(a) shows a sample figure of generated motion heatmap.

3.3 Feature Detection and Tracking Using ORB

In this step, we define a set of POIs for each input frame. We use the heatmap obtained in the previous step and use it as a mask to define the ROI. Then we use ORB (Oriented FAST and Rotated BRIEF) feature detector to define the POIs [22]. First, ORB uses FAST (Features from Accelerated Segment Test) to find key points [23].Then the top ***n*** points are selected using Harris Corner [24]. We use the intensity centroid [25] as a measure of corner orientation. A corner's intensity is assumed to be an offset from its center, and this vector is used to compute an orientation. The moments of the patch are defined as:

$$m_{ab} = \sum_{a,b} j^a, k^b I(j,k) \tag{5}$$

Then the centroid is found using the following formula,

$$c = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{6}$$

A vector is constructed from center of the detected corner ***O*** to centroid $\overrightarrow{OC}$, the orientation of the patch then becomes:

$$\theta = atan2(m_{01}, m_{10}) \tag{7}$$

Where, ***I(j, k)*** is the intensities of the image, ***atan2*** is the quadrant aware version of arctan. To improve rotation invariance we compute the moments while ***j*** and ***k*** remaining in a circular region of radius ***g***, which is also chosen to be the patch size so that ***j*** and ***k*** run from ***(-g, g)***. In our case, we defined ***n*** to be 500. Then BRIEF(Binary Robust Independent Elementary Features) descriptor is used for describing the identified key points [26], which gives a bit string description for an image patch constructed from a set of binary intensity tests. Let us assume ***u*** is a smoothed image, a binary test τ would be defined as equation (8):

$$\tau(u: j,k) = \begin{cases} 1: & u(j) < u(k) \\ 0: & u(j) \geq u(k) \end{cases} \quad (8)$$

Where ***u(j)*** is the intensity of ***u*** at point ***j***. The feature is defined as vector of ***t*** binary tests.

$$f_n(u) = \sum_{1 \leq i \leq t} 2^{i-1} \tau(u: j_i, k_i) \quad (9)$$

A Gaussian distribution around the center of the patch was used as test. ***t*** had a vector length of 256. ORB is rotation invariant and resistant to noise, while it is also efficient and fast compared to other feature descriptors. Being rotation invariant, it can detect the same features from different angles. Once we detect the POIs, we track them across multiple frames. We use Kanade-Lucas-Thomasi feature tracker for this purpose [27, 28]. The key points might change with time. New subjects might enter the frame, or old ones might exit the frame, which will cause our previous key points estimation to be inaccurate. To address this issue, we redefine the points of interests (POI) at a fixed interval ***k***.

Figure3(a) (a) (b)

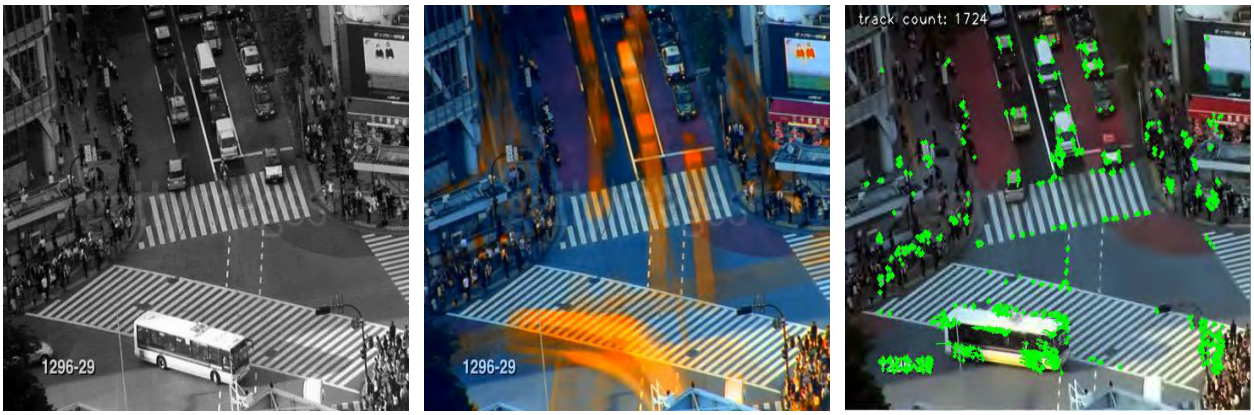


Figure3(b) (c) (d)

Figure 4: (a) Generated Motion Heatmap of the location of Figure 3(a), (b) Detected features in Figure 3(a), (c) Generated Motion Heatmap of the location of Figure 3(c), (d) Detected features in Figure 3(c)

After matching features between the frames, we get a set of vectors for each frame:

$$V = \{V_1, \ldots\ldots, V_n | V_i = (x_i, y_i, d_i, \theta_i, r_i)\} \quad (10)$$

Where, x_i and y_i are coordinates for feature i, d_i is the distance between the feature in the previous frame the matched feature in current frame, θ_i is the orientation or angle of the feature, it has a value between (0, 360) degrees, it is measured in relevance to the image coordinate system, that is in clockwise, r_i is the response by which the strongest key points has been selected. A few static and noise features, i.e. features that moved less than 2 pixels were removed in this step. Noise features have a big angular and magnitude difference with their nearest neighbors. Figure 4(b) shows some detected features from Figure 3(a).

3.4 Dividing the Frame into Uniform Cubes and Detecting the Motion Pattern within

Identifying the motion structure in an extremely crowded scenario is difficult, because a large number of completely independent activities occur in different part of the frame. The motion between different local areas can be generated by different subjects and can have different rate of changing. However, the motion of the entire frame, such as the motion heatmap we generated earlier, might not always accurately portray the motion patterns of separate independent events going in different parts of the frame.

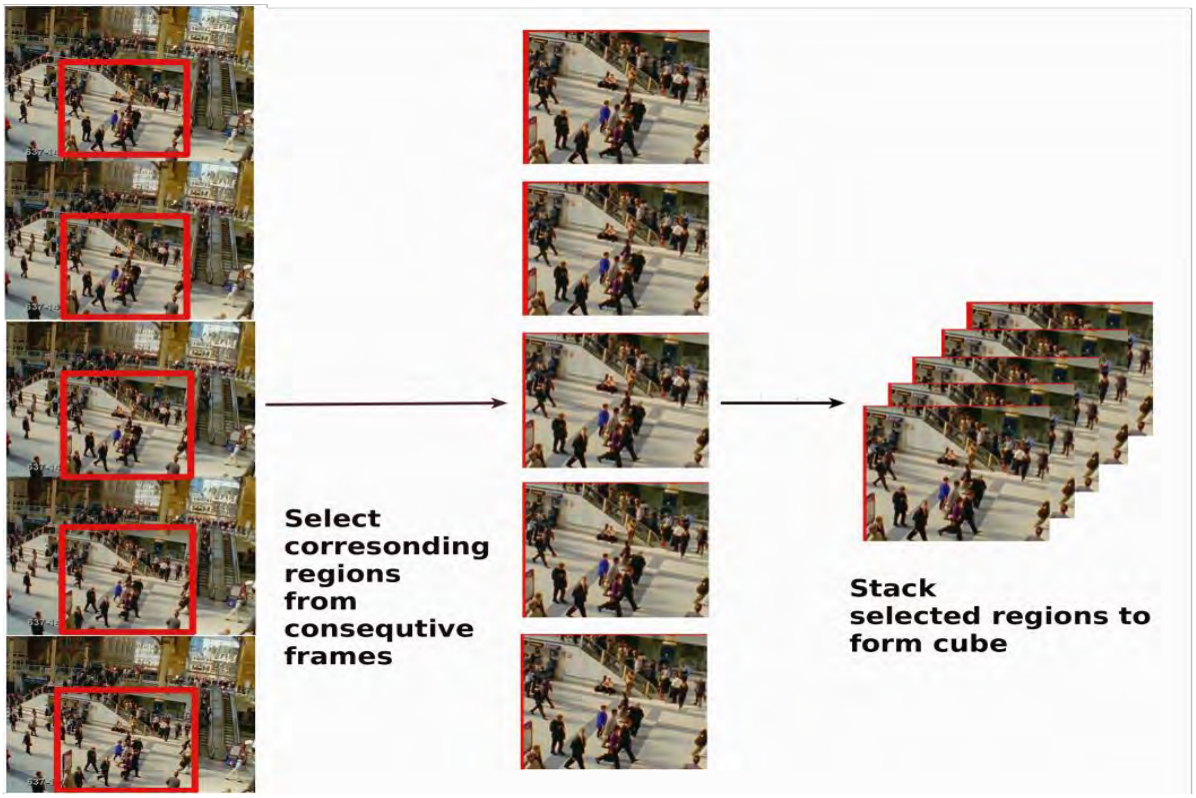


Figure 5: Spatio-temporal Cube Formation

We can isolate the local activities by dividing the video into local spatial-temporal volumes of fixed size. Then we identify a compact motion pattern for each volume, we capture the motion structure of the video [29]. For each input frame, we scale it to three sizes and divide every scaled frame in equal and non-overlapping regions. Corresponding regions in consecutive frames are

stacked to form a 3D cube, which is shown in Figure 5. For every k frames, we identify non overlapping regions, and for every pixel p in a non-overlapping region R, we calculate the spatial-temporal gradient R_p using Sobel derivative function.

$$R_p = [R_{p,v}, R_{p,w}, R_{p,z}]^T = \left[\frac{\partial R}{\partial v}, \frac{\partial R}{\partial w}, \frac{\partial R}{\partial z}\right]^T \quad (11)$$

Where, v, w, z are the horizontal, vertical and temporal dimensions of the video respectively. The collection of spatial-temporal gradient of every pixel within a region R forms the dominant motion pattern within R. We model the gradient distribution as a three dimensional Gaussian distribution $G(\alpha, \beta)$; where,

$$\alpha = \frac{1}{N}\sum_{1}^{N} R_p \quad (12)$$

$$\beta = \frac{1}{N}\sum_{1}^{N} (R_p - \alpha)(R_p - \alpha)^T \quad (13)$$

Where N is the total number of pixels. After this we have a four dimensional feature vector containing both the set of feature vectors and three dimensional gradient features for each pixel of the video, which we use later for classification purposes.

After combining the detected features and underlying motion structure, we can define the final feature vector as:

$$S = \{V, G(\alpha, \beta)\} \quad (14)$$

3.5 Defining a Criteria to Separate Normal and Abnormal Events

The specific threshold is determined based on comparison between the identified feature vectors and 3D gradient features in both abnormal and normal scenarios, which can be used in later cases in identification of abnormal events. The threshold might change based on camera position, time of the day, location etc. So special configuration might be necessary to determine the threshold. After we deploy the model, every new captured footage once processed and checked for abnormalities, can be used for further learning thus increasing the accuracy with time.

Figure 6: Some Detected Abnormal Events

CHAPTER 04

EXPERIMENTAL ANALYSIS

4.1 Setup

We used Web Dataset [30], UMN Dataset [31], UCSD Anomaly Dataset [32] for testing our model. These datasets contain various normal and abnormal events in both indoor and outdoor scenes in various times of the day. The events are both temporal and spatial as well as both local and global.

All experiments were conducted in a personal computer with Intel Core i5 2.80 GHz CPU with 16 gigabytes of RAM with 64 bit Ubuntu 14.04 as operating system.

4.2 Datasets

4.2.1 UCSD Dataset

The UCSD dataset was acquired from a stationary camera situated at a high place above pedestrian walkways. It is divided into 2 parts, Peds1 where the people are walking to and away from the camera, and Peds2 where the camera is situated in perpendicular position from the walkway so the crowd movement is parallel to the camera plane.

Peds1 has 34 training samples and 36 training samples, and Peds2 has 16 training samples and 12 testing samples. Abnormal events are defined as sudden change in crowd motion patterns or entities which are not usual pedestrians such as bikers, skaters or people in wheelchairs coming onto the walkways. Figure 7 has some examples of scenes from UCSD dataset with identified features.

Figure 7: Figures from UCSD Dataset

4.2.2 UMN Dataset

The UMN dataset is a dataset from University of Minnesota. It consists of videos of resolution 320 x 240 with 11 different scenarios from three different scenes from both indoors and outdoors. The abnormal event being a crowd suddenly running and people dispersing in different directions.

Figure 8 shows some sample frames from the UMN dataset with identified features.

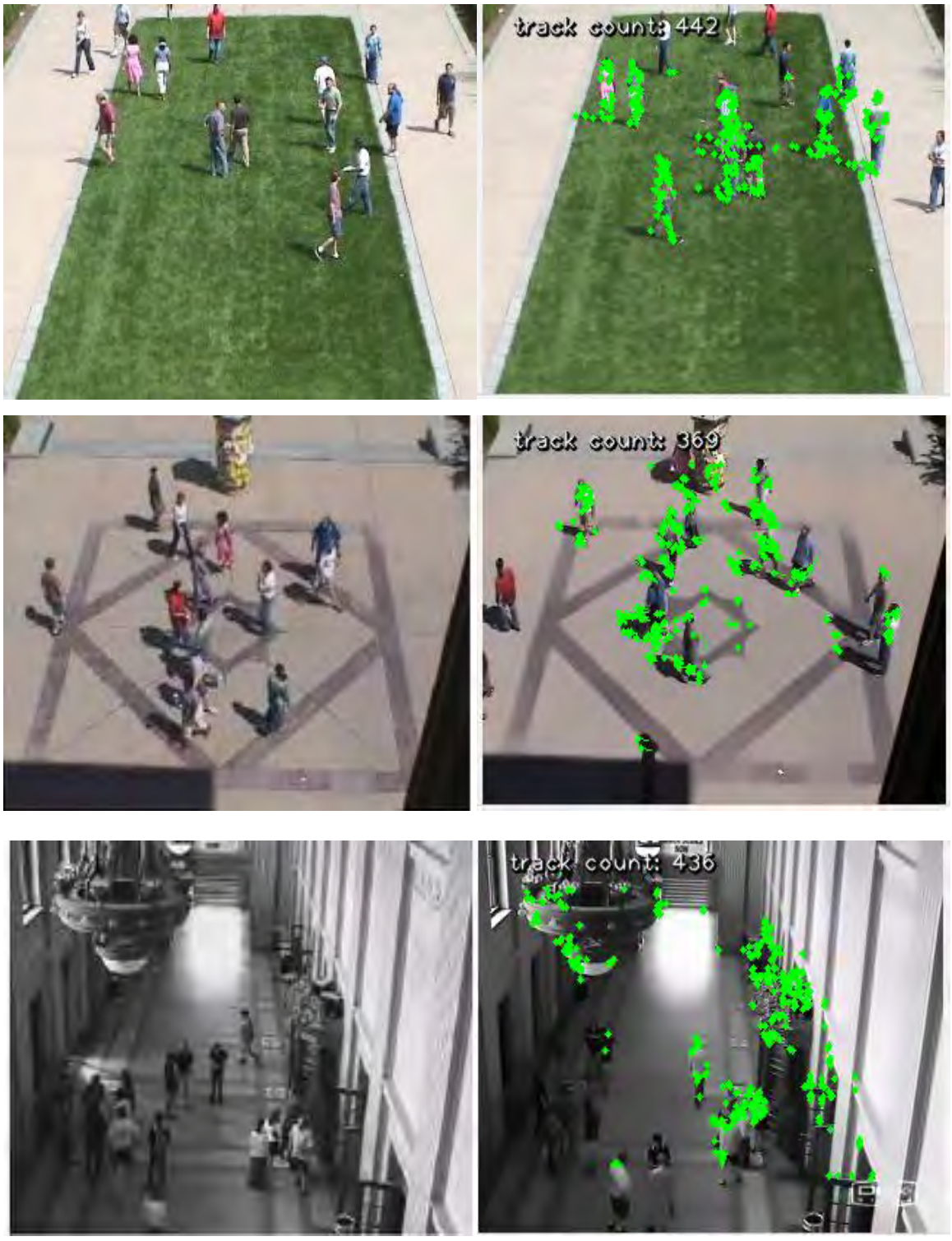


Fig 8: Figures from UMN dataset

4.2.3 Web Dataset

The Web dataset consists of a set of videos from different urban scenarios. Twelve scenarios contain normal events such as pedestrians crossing streets, people using escalators and eight scenarios contain abnormal situations like clashing crowds or escape panics.

Figure 8 contains some sample pictures from this dataset with identified features.

Figure 9: Figures from Web Dataset

4.3 Feature Vector Generation

We create a combined data file for all of the datasets containing the feature vectors of all of the frames for videos of the datasets. For each video, frames are extracted and then resized to three scales 20 x 20, 30 x 40 and 120 x 160. Then we divide the frames into 10 x 10 non overlapping regions. And corresponding patches in 5 overlapping frames are then used to form 10 x 10 x 5 3D cubes. For each cube, 3D gradient feature vector is generated. Combined with the features detected using ORB detector, we have a set of 3D feature vectors of shape [2, 2000, 1500]. If in any case the number of detected features is not uniform, zeroes are used to make the size uniform.

4.4 Classifiers Setup

50% of the generated features is used for training and 50% for testing purposes. We train 6 classifiers on the given data and then compare their performances.

4.4.1 Support Vector Machine (SVM)

Support vector machines are a kind of supervised learning model with associated learning algorithms. SVMs analyze data used for classification. Given a set of training data, each belonging

to either one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. A SVM constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance or functional margin to the nearest training data point of any class, since in general it is observed that the larger the margin the lower the generalization error of the classifier. A normal SVM and a polynomial support vector machine of degree 2 is used on our data for supervised learning. Given labeled training data, this algorithm gives a separating hyper plane which is used to classify new examples based on which side of the hyper plane the new data falls. Since this model can not take 3D values, we had to flatten our data to a 2D shape [100, 60000].

4.4.2 Logistics Regression

Logistics Regression is a binary classifier for problems with binary dependent variables that is variables with only two possible values. We use logistics regression model to evaluate the performance of our model. We used gradient descent optimizer for classification with predefined learning rate of 0.1 and training epochs 10. Logistics Regression performs admirably well for problems with binary solutions. For this model we also used the flattened version of our data.

4.4.3 Naive Bayes Classifier

Naive Bayes classifier is a probabilistic classifier based on Bayes Theorem which assumes that the features are strongly independent of each other that is the value of one variable does not depend

on that of other. We used a Guassian Naïve Bayes Method to classify our data. For this classifier we had to use the flattened version of data as well.

4.4.4 Neural Network

Neural networks are often used to classify large amount of data. We use a fully connected neural network with three layers. The network weights are small random numbers between 0 to 0.05 generated from a uniform distribution. Rectifier activation function is used in the first two layers, but on the last layer we use sigmoid activation function. This is because sigmoid ensures that our network output is between 0 to 1 and easy to map. Since our problem is a binary classification problem we use logarithmic loss as a loss function and efficient gradient descent algorithm [33] for optimizer. Our model does 150 iterations with a batch size of 10.

4.4.5 Convolutional Neural Network

Convolutional Neural Networks are a specific type of neural network which specialize in image classification. For our model we use a 2D convolutional neural network, with 1 x 1 shape and 64 output filters. On top of that we apply another 1 x 1 convolution with 32 output filters. We use rectifier activation function in first 2 layers and for the last layer we use softmax activation.

We use sparse categorical cross entropy as our loss function efficient gradient descent algorithm for optimizer. We use the same amount of iterations and batch size as neural networks for our CNN.

4.5 Comparing the results

Performance of the proposed model for different datasets is depicted in Table 2. The proposed model shows steady performance for all datasets and achieves an average accuracy of 96.12% for all datasets.

Table 2: Accuracy of Classification for Different Datasets

Dataset	Accuracy (%)
Web Dataset	97.34
UCSD Dataset	94.28
UMN Dataset	96.74

We tested our model with various classifiers and found out that CNN shows the best accuracy.

Figure 10 shows the comparison among the accuracies in a bar chart form.

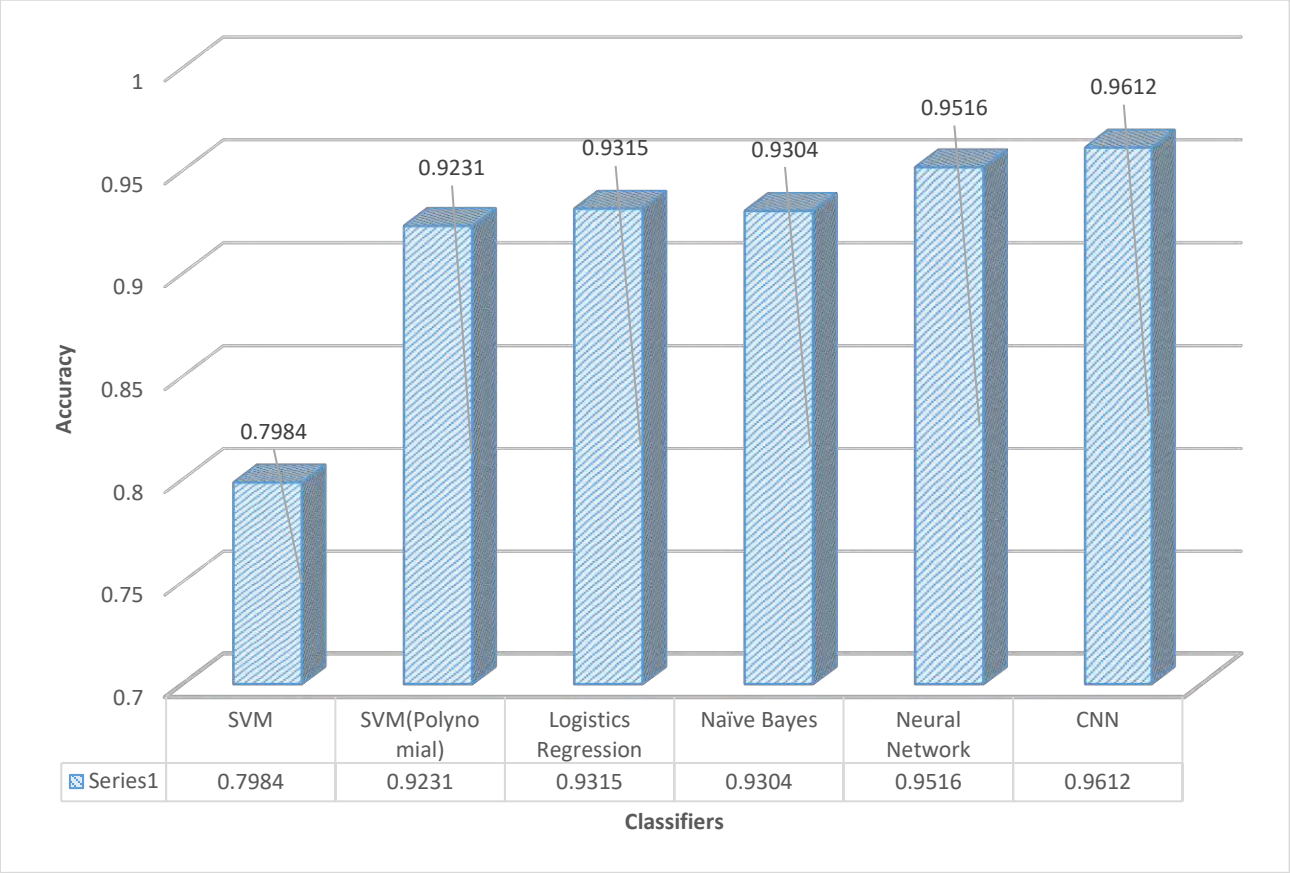


Figure 10: Comparison of performance among the classifiers

As we can see the performance for Support Vector Machine improved dramatically after using a polynomial model with degree 2. Logistics Regression achieved high accuracy with our model because it is a binary classification problem. Naïve Bayes classifier also performed well. Neural network also performed very well on our model, since the data is large. CNN improves upon the result of Neural Network classifier. Because CNN performs better with increased number of data and it specializes in classifying image data. CNN allows the networks to have fewer weights as these parameters are shared between neurons and it uses convolutions to analyze images.

CHAPTER 05

CONCLUSIONS AND FUTURE WORKS

5.1 Concluding Remarks

In this paper we propose a pattern recognition method to determine abnormal situations in crowded scenarios. We proposed a model which can successfully identify both local and global abnormal models in temporal and spatial scenarios. The model is self-sufficient as once it is deployed with preliminary training it can further increase its accuracy by using the captured footages for learning as well. Our model calculates a motion heatmap of the region which is later used as mask for detecting and tracking features across multiple frames, while also dividing the video frame into multiple non-overlapping regions and calculating motion pattern within each separate region. Our model shows significant improvement over the previously proposed models in accuracy. It is also portable meaning it can be deployed in any situation with just a few days prior training to detect events, because it trains itself with time.

5.2 Future Works

The potential future directions for research based on the results presented in this thesis can be characterized into the following sections.

5.2.1 Classifying Abnormal Events

We would like to explore the possibilities of detecting an abnormal event and at the same time also classify them into categories based on severity or necessity of immediate response.

5.2.2 Using GPU to Decrease Computation Time

Although our model can identify abnormal event from input within acceptable time limit, the

training period is still very large. It's because we have to process a large amount of input data to properly train our model. We will try to decrease the training time utilising GPU resources and make the training period shorter.

REFERENCES

1. Yen, S., & Wang, C. (2013). Abnormal Event Detection Using HOSF. 2013 International Conference on IT Convergence and Security (ICITCS).
2. Cong, Y., Yuan, J., & Liu, J. (2013). Abnormal event detection in crowded scenes using sparse representation. Pattern Recognition,46(7), 1851-1864.
3. Avidan, S. (2005). Ensemble Tracking. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 261-271.
4. Cong, Y., Gong, H., Zhu, S., & Tang, Y. (2009). Flow mosaicking: Real-time pedestrian counting without scene-specific learning. 2009 IEEE Conference on Computer Vision and Pattern Recognition.
5. Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 886-893.
6. Stauffer, C., & Grimson, W. (1999). Adaptive background mixture models for real-time tracking. Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149).
7. Yuan, J., Liu, Z., & Wu, Y. (2009). Discriminative subvolume search for efficient action detection. 2009 IEEE Conference on Computer Vision and Pattern Recognition.
8. Ma, R., Li, L., Huang, W., &Tian, Q. (2004). On pixel count based crowd density estimation for visual surveillance. IEEE Conference on Cybernetics and Intelligent Systems, 2004., 170-173.

9. Marana, A. (1997). Estimation of crowd density using image processing. IEE Colloquium on Image Processing for Security Applications.
10. Rahmalan, H., Nixon, M., & Carter, J. (2006). On crowd density estimation for surveillance. IET Conference on Crime and Security.
11. V., J., & S. (2003). Detecting pedestrians using patterns of motion and appearance. Proceedings Ninth IEEE International Conference on Computer Vision.
12. Mehran, R., Oyama, A., & Shah, M. (2009). Abnormal crowd behavior detection using social force model. 2009 IEEE Conference on Computer Vision and Pattern Recognition.
13. Zhong, H., Shi, J., &Visontai, M. (n.d.). Detecting unusual activity in video. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.
14. Benezeth, Y., Jodoin, P., Saligrama, V., & Rosenberger, C. (2009). Abnormal events detection based on spatio-temporal co-occurences. 2009 IEEE Conference on Computer Vision and Pattern Recognition.
15. Ihaddadene, N., &Djeraba, C. (2008). Real-time crowd motion analysis. 2008 19th International Conference on Pattern Recognition.
16. Ali, S., & Shah, M. (2007). A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis. 2007 IEEE Conference on Computer Vision and Pattern Recognition.
17. Andrade, E., Blunsden, S., & Fisher, R. (2006). Hidden Markov Models for Optical Flow Analysis in Crowds. 18th International Conference on Pattern Recognition (ICPR'06).
18. Andrade, E., Blunsden, S., & Fisher, R. (2006). Modelling Crowd Scenes for Event Detection. 18th International Conference on Pattern Recognition (ICPR'06).

19. Buades, A., Coll, B., & Morel, J. (2011). Non-Local Means Denoising. Image Processing On Line,1.
20. Zivkovic, Z. (2004). Improved adaptive Gaussian mixture model for background subtraction. Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.
21. Zivkovic, Z., &Heijden, F. V. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters,27(7), 773-780.
22. Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. 2011 International Conference on Computer Vision.
23. Rosten, E., & Drummond, T. (2006). Machine Learning for High-Speed Corner Detection. Computer Vision – ECCV 2006 Lecture Notes in Computer Science, 430-443.
24. Harris, C., & Stephens, M. (1988). A Combined Corner and Edge Detector. Procedings of the Alvey Vision Conference 1988.
25. Rosin, P. L. (1999). Measuring Corner Properties. Computer Vision and Image Understanding, 73(2), 291-307.
26. Calonder, M., Lepetit, V., Strecha, C., &Fua, P. (2010). BRIEF: Binary Robust Independent Elementary Features. Computer Vision – ECCV 2010 Lecture Notes in Computer Science, 778-792.
27. Shi, J., &Tomasi, C. (1994). Good features to track. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94.
28. Lucas, B. D., &Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. International Joint Conference on Artificial Intelligenc, 674-679.

29. Kratz, L., & Nishino, K. (2009). Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. 2009 IEEE Conference on Computer Vision and Pattern Recognition.
30. Web dataset of unusual crowd activity, available at http://crcv.ucf.edu/projects/ Abnormal_Crowd/Normal_Abnormal_Crowd.zip
31. Unusual crowd activity dataset of University of Minnesota, available from http://mha.cs.umn.edu/movies/crowd-activity-all.avi.
32. UCSD anomaly dataset available at http://www.svcl.ucsd.edu/projects/anomaly /UCSD_Anomaly_Dataset.tar.gz
33. P.Kingma, D., & Ba, J. L. (2015). Adam: A Method For Stochastic Optimization. ICLR