



JAVERIA ZAFAR

**01-235191-100**

AROoba MALIK

**01-235191-005**

# **Abnormal Event Detection**

**Bachelor of Science in Information Technology**

Supervisor: Mahwish Pervaiz

Department of Computer Science  
Bahria University, Islamabad

December 30<sup>th</sup>, 2022

# Certificate

We accept the work contained in the report titled “Abnormal Event Detection”, written by Javeria Zafar and Arooba Malik as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by . . . :

Supervisor: Mahwish Pervaiz

---

Internal Examiner: Faisal Asad ur Rehman

---

External Examiner: Muhammad Asif

---

Project Coordinator: Zubaria Inayat

---

Head of the Department: Dr Arif ur Rehman

---

December 30<sup>th</sup>, 2022

# Abstract

Abnormal event detection is one of the foremost important task in analysis applications. because the traditional and abnormal events have some similarities, a lot of discriminating ways or motion data ought to be explored. quick abnormal event detection meets the growing demand to methodology a large style of security videos. For that purpose we have proposed a model that used for event detection. System detect the event whether it's normal or abnormal on basis of movement and velocity. For human detection we use HOG descriptor. For classification we used five different classifiers: decision tree, naïve Bayes, bagging, linear SVC and random forest. For evaluating the performance of our model we have a tendency to used 2 datasets: Avenue and Web dataset. Results shows that our model offers sensible accuracy and shows enhancement.

# Acknowledgments

This project would not have reached its desired outcome without the help of countless people but firstly, we pay our due regards to Allah Almighty who gave us the capability and knowledge to fully comprehend the project, His blessings are countless and we are forever grateful to Him. We would also like to appreciate the efforts of our project supervisor Ma'am Mahwish Pervaiz who helped us through every obstacle and hurdle that we faced in our project. She was supportive, her influence was mighty and highly effective. Lastly, we would like to mention all the teachers who taught us through our whole degree duration.

JAVERIA ZAFAR, AROOBA MALIK  
Bahira University Islamabad, Pakistan

December 30<sup>th</sup>, 2022

*“Out of clutter, find Simplicity. From discord, find Harmony. In the middle of difficulty lies Opportunity.”*

Albert Einstein

# Contents

<b>Abstract</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Challenges . . . . .	2
1.2 Objective . . . . .	3
1.3 Problem Description . . . . .	3
1.4 Project Scope . . . . .	3
1.5 Methodology . . . . .	4
1.6 Feasibility Study . . . . .	5
1.6.1 Risks Involved . . . . .	5
1.6.2 Resource Requirement . . . . .	5
1.6.3 Solution Application Areas . . . . .	5
1.7 Tools/Technology . . . . .	6
1.7.1 Tools . . . . .	6
1.7.2 Language . . . . .	6
1.7.3 Libraries . . . . .	6
<b>2 Literature Review</b>	<b>7</b>
<b>3 Requirement Specifications</b>	<b>13</b>
3.1 Existing Systems . . . . .	13
3.2 Proposed System . . . . .	13
3.3 Require Specification . . . . .	14
3.3.1 Functional Requirements . . . . .	14
3.4 Non-Functional Requirements . . . . .	16
3.4.1 Availability . . . . .	16
3.4.2 Reliability . . . . .	16
3.4.3 Usability . . . . .	16
3.4.4 Security . . . . .	16
3.4.5 User Friendly: . . . . .	17
3.5 Use Case Diagram . . . . .	17
3.5.1 Sub-Use Case Diagrams . . . . .	18
3.6 Use Case Dictionary . . . . .	19
3.6.1 Import Video . . . . .	19
3.6.2 Pre-Processing . . . . .	19
3.6.3 Object Detection . . . . .	20
3.6.4 Human Verification . . . . .	21

3.6.5	Feature Extraction . . . . .	21
3.6.6	Human Activity Recognition . . . . .	22
<b>4</b>	<b>Design</b>	<b>24</b>
4.1	System Architecture . . . . .	24
4.2	Design Constraints . . . . .	25
4.3	Design Methodology . . . . .	25
4.4	Sequence Diagram . . . . .	27
4.4.1	Human Detection . . . . .	27
4.4.2	Activity Recognition . . . . .	27
4.5	GUI Design . . . . .	28
4.5.1	User Interface 1 . . . . .	28
4.5.2	User Interface 2 . . . . .	28
4.5.3	User Interface 3 . . . . .	29
4.5.4	User Interface 4 . . . . .	29
<b>5</b>	<b>System Implementation</b>	<b>30</b>
5.1	Internal Component . . . . .	30
5.2	Functionality of Components . . . . .	31
5.2.1	Data Set . . . . .	31
5.2.2	Pre-processing . . . . .	32
5.2.3	Object Detection . . . . .	33
5.2.4	Human Verification . . . . .	33
5.2.5	Feature Extraction . . . . .	35
5.2.6	Classification . . . . .	36
5.2.7	Accuracy Generated From Model . . . . .	45
5.3	Libraries . . . . .	45
5.3.1	OS . . . . .	45
5.3.2	Numpy . . . . .	45
5.3.3	Pandas . . . . .	46
5.3.4	CV2 . . . . .	46
5.3.5	Glob . . . . .	46
5.3.6	Scikit-learn . . . . .	46
<b>6</b>	<b>System Testing and Evaluation</b>	<b>47</b>
6.1	Graphical User Interface Testing . . . . .	47
6.1.1	Video Input . . . . .	47
6.1.2	Video Display . . . . .	48
6.1.3	Anomaly Detection Alert . . . . .	49
6.2	Usability Testing . . . . .	51
6.3	Test Cases . . . . .	51
6.3.1	Model Training . . . . .	51
6.3.2	Model Testing . . . . .	52
6.3.3	Video Input . . . . .	52
6.3.4	Load Multiple Video . . . . .	53
6.3.5	Frame Extraction . . . . .	53
6.3.6	Processing . . . . .	54

6.3.7	Video Display . . . . .	54
6.3.8	Anomaly Alert . . . . .	55
6.4	Exceptional Handling . . . . .	55
<b>7</b>	<b>Conclusions</b>	<b>56</b>
	<b>References</b>	<b>57</b>



# List of Figures

1.1	Web dataset . . . . .	4
1.2	Avenue dataset . . . . .	4
1.3	Methodology Flowchart . . . . .	5
3.1	Use Case Diagram . . . . .	17
3.2	Use Case For Video Import . . . . .	18
3.3	Use Case For Video Display . . . . .	18
3.4	Use Case For Anomaly Alert . . . . .	18
4.1	System Architecture . . . . .	25
4.2	Methodology Steps . . . . .	26
4.3	Human Detection . . . . .	27
4.4	Human Activity Recognition . . . . .	27
4.5	User Interface 1 . . . . .	28
4.6	User Interface 2 . . . . .	28
4.7	User Interface 3 . . . . .	29
4.8	User Interface 4 . . . . .	29
5.1	Avenue Dataset . . . . .	31
5.2	Web Dataset . . . . .	31
5.3	Video Input . . . . .	32
5.4	Histogram of oriented gradient . . . . .	35
5.5	Feature Extraction . . . . .	35
5.6	Classification . . . . .	37
5.7	Accuracy . . . . .	45
6.1	Load Image . . . . .	48
6.2	Without Video . . . . .	48
6.3	With Video . . . . .	49
6.4	Detection Alert . . . . .	49
6.5	Detection Alert . . . . .	50
6.6	Detection Alert . . . . .	50
6.7	Detection Alert . . . . .	51

# List of Tables

3.1	Use Case Dictionary for Import Video . . . . .	19
3.2	Use Case Dictionary for Pre-Processing . . . . .	20
3.3	Use Case Dictionary for Object Detection . . . . .	20
3.4	Use Case Dictionary for Human Verification . . . . .	21
3.5	Use Case Dictionary for Feature Extraction . . . . .	22
3.6	Use Case Dictionary for Human Activity Recognition . . . . .	23
5.1	Accuracy Results . . . . .	45
6.1	Test Case for Train Model . . . . .	52
6.2	Test Case for Test Model . . . . .	52
6.3	Test Case for Video Input . . . . .	53
6.4	Test Case for Load Multiple Videos . . . . .	53
6.5	Test Case for Frame Extraction . . . . .	54
6.6	Test Case for Processing . . . . .	54
6.7	Test Case for Video Display . . . . .	55
6.8	Test Case for Anomaly Alert . . . . .	55

# Acronyms and Abbreviations

OVR	Occurrence/Variance Report
MII	Motion Information Image
SVC	Scalable video coding
UMN	University of Minnesota
LSTM	Long Short-Term Memory
NCS	Neural Compute Stick
MIL	Multiple Instance Learning
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
NFR	Non-Functional Requirements
HTML	HyperText Markup Language
CSS	Cascading Style Sheet

# Chapter 1

## Introduction

Event recognition is a sub-field of computer vision which identifies input videos with the objective of determining anomalous event when arise. Event detection is a technique for analysing event streams to find groups of events that are abnormally created. Depends on the context we all agree that abnormal events are unexpected events that happens less often than normal events.

Among the newest research problems in the video field, abnormal event detection perceives the specific frames containing an anomaly. Abnormal event detection has increased impact on national security and people's livelihoods. One of the main areas that individuals started utilising for security reasons and activity monitoring is video surveillance. As society becomes more modernised, more surveillance cameras are being installed in public spaces like parks and shopping centres, providing a massive volume of video every second that is difficult for a person to process and identify any unusual events in.

In [1] a system of abnormal event detection in videos using spatiotemporal autoencoder was proposed in which spatiotemporal architecture was presented. This study includes two main parts, one for representing spatial feature and the other for learning how to evaluate spatial features over time. They used three datasets including Avenue, Subway and UCSD in their study. Experimental results in these benchmarks shows that the detection accuracy of their method was comparable to state-of-the-art methods at considerable speed of up to 140 frames per second.

In [2] a method was proposed in which it was suggested to use a feasible automated video analytic system to count and monitor crowds. This system makes use of a few key approaches, such as a video processing back-end that incorporates tracking and human subject detection as well as a front-end graphical interface for operators who use

conventional and CNN-based object recognition algorithms. This technology is able to accept many video streams from a central storage location and reduce human workload.

In [3] a way to identify abnormal events in human group activities was suggested by a system of abnormal event detection on the BMTT-PETS 2017 surveillance challenge. Their primary contribution was the creation of a method that uses supervised learning and action isolation to learn from a small number of videos. To eliminate unwanted static visual elements, the background was subtracted from the initial image. They then employ an LSTM network for sequence learning after using a CNN for feature extraction. The LSTM network receives as an input the output of CNN's final fully linked layer. They finally obtain the classification scores using a linear SVM. Their approach highlights how non-deep learning based techniques such as linear SVM and background subtraction can be combined with deep learning techniques such as LSTMs and CNN's to build machine learning systems with fewer data.

In [4] a method was proposed in which novel technique for anomalous crowd event detection in surveillance videos was used for unsupervised anomaly detection and localization based on deep spatiotemporal translation networks. The suggested method was built upon a fresh MII model that was created utilising optical flow. Optical flow vectors that are able to create images with motion information that are then taught and tested using CNN.

## 1.1 Challenges

An appropriate feature extraction, defining normal behaviours, handling an imbalanced distribution of normal and abnormal data, addressing variations in abnormal behaviour, sparse occurrence of abnormal events, environmental variations, and camera movements are some of the challenges in anomaly detection from the existing systems. Additionally, we could see that the present object recognition and tracking technologies do not place facial identification at the top for improved surveillance capabilities. It also appears challenging to discover anomalies other than motion abnormalities given the need for enormous datasets to train the algorithm perfectly. Also, the activity must be determined by the individual's size. Small images are more challenging to detect.

## 1.2 Objective

To design a system that automates the intimation of anomalies happened in a particular place with the help of a video surveillance camera which does not require human involvement to perform the action.

## 1.3 Problem Description

The reason for developing this project is that the task of abnormal event detection in computer vision is challenging. Due to the massive increase in population we want to avoid any anomaly and unwanted condition and for that the need for accurate and significant applications has been increased. Surveillance cameras are used every where in public places like parks, malls and markets to recognise any unusual occurrence that can inconvenience to people. For that we need monitor rooms and a lot of people to look at those monitors, so some time if there is an anomaly happened and no person is watching they can miss that. So this system became helpful for abnormal event detection as it does not require the involvement of human to detect the abnormal event yet it alerts the authorities whenever any anomaly is happened.

Many pattern learning techniques have been created in the last ten years, and the majority of them make the assumption that any pattern that deviates from this typical pattern should be abnormal. Because the normal pattern may include an excessive number of various events and actions, it is actually impossible to provide a technique to identify the entire range of normal patterns. Since abnormality events may also involve numerous comparable events and behaviours, defining abnormality events can be difficult.

## 1.4 Project Scope

The main target of this project is small to medium density crowd. Our system will used in places like Park, Malls and Market. Specific size of images or individuals required to detect the activity. The project employs a trained model that has been taught to analyze video input, perform model inference, and display output if an anomaly is discovered. For that purpose we used web and avenue dataset.



Figure 1.1: Web dataset

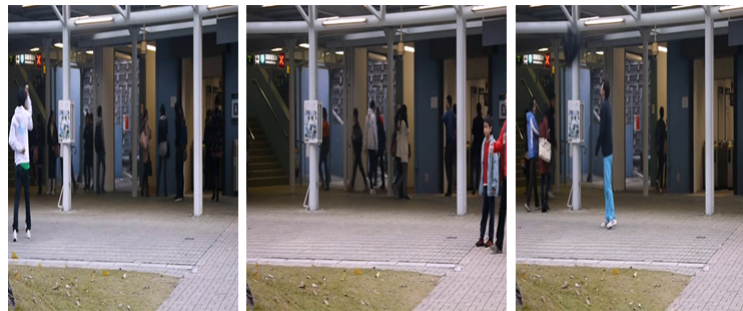


Figure 1.2: Avenue dataset

## 1.5 Methodology

This section provides the approach used for identifying and classification using machine learning. The method is understood by the workflow model provided in figure [1]. We used two different datasets in this project that includes web dataset and avenue dataset. The web dataset includes 12 videos of regular pedestrians and 8 videos of actual fights, clashes, and escapes from fear. The avenue dataset includes 16 training and 21 testing video clips and it was captured in Chinese University of Hong Kong (CUHK). It contains 15328 training, 15324 testing while 30652 frames in total.

Firstly in order to train the model with the video clips, we had selected the appropriate dataset. Then import the video, read the content, extract the frames from the video input, and save them as images in the temporary memory for further processing to carry out the data preparation. The system fixes a size for each frame. The fixed height and width for each frame size is usually 128 for making the frames more sharp, while the noise and objects in the background is removed and detect the object. After this phase, the system verifies the human if it is a human or other object. After the verification process, the extra features are extracted on the basis of orientation, velocity and movement. In the end, the system pass the features for classification and the output is generated.

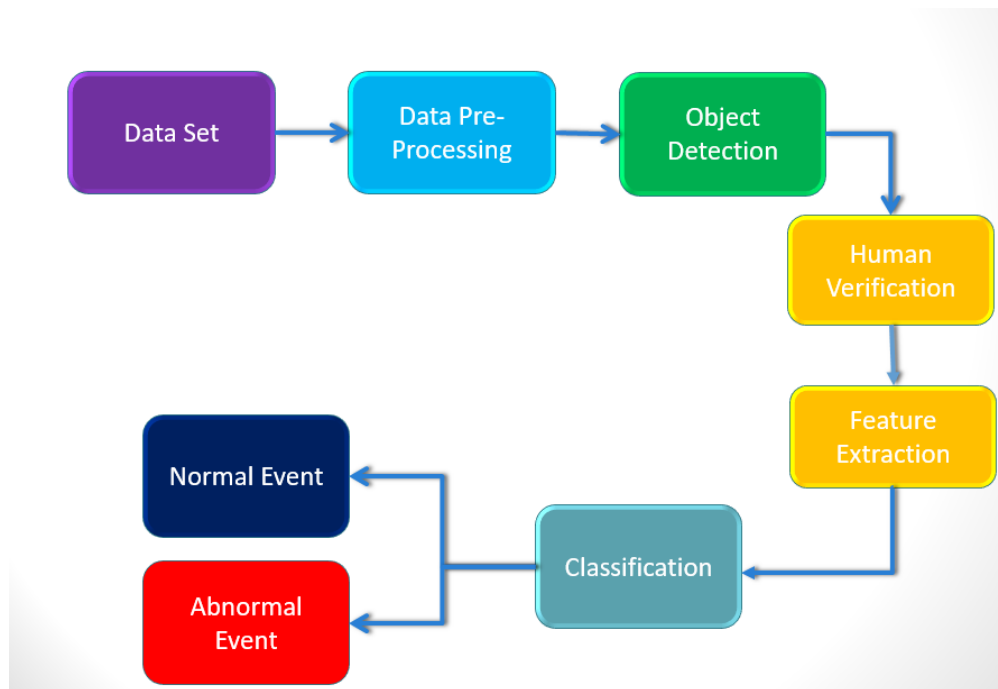


Figure 1.3: Methodology Flowchart

## 1.6 Feasibility Study

We tried our best to meet the schedule of our project as we worked throughout our semesters and used the best technology for our project because we want to use this project for future use as well. We used simple languages and logic so it became easy for other users to understand and make it more feasible.

### 1.6.1 Risks Involved

- Large crowd is not targeted.
- Specific size of images or individuals required to detect the activity

### 1.6.2 Resource Requirement

- Camera
- An interface for the system to integrate properly

### 1.6.3 Solution Application Areas

Our project will be used in

- Shopping malls



- Parks
- Markets

## **1.7 Tools/Technology**

### **1.7.1 Tools**

- Jupyter Notebook
- Visual Studio Code

### **1.7.2 Language**

- Python
- HTML
- CSS

### **1.7.3 Libraries**

- Numpy
- OpenCV
- Pandas
- OS

## Chapter 2

# Literature Review

A literature review is a piece of writing that aids in connecting the understanding of the subject matter that we have selected for the final year project. It is crucial that we research all relevant aspects of the chosen issue and that we are capable of responding to queries about the research. Experts believe that someone can accomplish this goal by conducting a literature review. Additionally, the literature review leads you to a thorough analysis of the particular research issue. Understanding the connected work is essential for any study in order to identify the flaws in the current system and formulate a new approach to the problem's solution that takes into account the problems found in the related work.

In [5] As one of the most important tasks in research applications, writers conducted research on the identification of anomalous events in crowded situations. The authors used de-noising input frames in this case. To lessen the blur and noise in the frames, they employed pixel-wise non-local Means (NLM) denoising. The gaussian mixture-based background and foreground segmentation algorithm is used in their model. The model selects the appropriate number of gaussian distributions for each pixel, offering greater adaptability for scenarios that may change owing to brightness and other factors. Additionally, they employ the motion heatmap creation, a two-dimensional histogram that identifies the regions in the video that underwent any motion movement at all. They used three datasets, the Web dataset, the UMN dataset, and the UCSD dataset, which each contain a variety of normal and aberrant occurrences, to evaluate the performance of the model they had suggested. They were 96.12 percent accurate across all datasets. As a result, the model's accuracy is improved over that of the earlier models that were suggested.

In [6] As it becomes more and more crucial in research applications, writers have examined abnormal event detection from movies using a two-Stream recurrent variational autoencoder. Unsupervised learning is used with a variational autoencoder to model complex distributions. It was a procreative model using neural network parameters. The authors analyse the long-term dependencies of a video sequence using Long Short Term Memory. However, it is well recognised that LSTM has limitations in how it expresses spatial information, which is essential to high-level visual semantics. The convolutional LSTM, in which all the state-to-state evolutions of memory cells are convolutional operations, is used by the authors to address the issue. The model is able to preserve spatial information while also capturing temporal dependencies. We can sum up by saying that the authors' contributions produce a semi-supervised result for abnormal event identification in videos. On multiple datasets, including AVENUE, UCSD, and SUBWAY, this method outperforms current top methods in terms of assessments at the frame, event, and pixel levels.

In [7] Authors investigated the use of feature coding for anomalous event detection in indoor video. They present a base lexicon that may be updated online to reflect changes in the video's context. The method that was proposed improved on traditional sparse coding, whose bases are stored permanently in the dictionary and are not changed after training. They are not suitable for unsupervised methods as a result. The spatio-temporal patch approach is utilised for video representation. Events are properly represented by a high dimensional feature vector. The use of high-dimensional feature sparse representation of events and model complexity create a difficult problem. Consequently, it is challenging to fit a reliable Gaussian model to the sparse data. When there are few training data and high dimensional features, sparse representation is the optimal method for abnormal event identification. The scientists employed two readily available video datasets from the entrance and exit surveillance cameras of subway stations for their experimental findings. Their method aids in reducing the quantity of false alarms at subway station departure and entrance gates. Compared to other accessible ways that are suggested by other researchers, their system was able to deliver more accurate results in a series of anomalous incidents.

In [8] With spatio-temporal adversarial networks that learn the spatio-temporal characteristics of normal patterns, the author suggested a unique abnormal event detection method. The spatio-temporal discriminator and the spatio-temporal generator are the two networks that make up the spatio-temporal adversarial networks. A clean frame for the typical scene and a deformed frame for the abnormal scenario were to be produced by the suggested generator, respectively. The spatio-temporal characteristics of typical patterns are well encoded using the bidirectional design. As a result, scenarios with anomalous spatiotemporal patterns can be quickly identified. The suggested discriminator has two functions. First, the discriminator aids the generator in learning the spatiotemporal characteristics of typical patterns during adversarial learning. The discriminator alone can identify aberrant events as the second step following adversarial learning. They used the UCSD Ped1, UCSD Ped2, and Avenue datasets for their studies and findings. The training sets of these datasets contain only normal events while the testing sets contain both normal and aberrant occurrences. Two evaluation indicators were used to assess the effectiveness of their suggested strategy. First, they used the area under curves-based frame-level evaluation. They also used event-level evaluation depending on the quantity of occurrences found. Comparing the findings of the suggested technique to those of the previous event-level evaluation strategies, it showed higher accuracy.

In [9] Authors suggested using a sparse reconstruction over the usual basis to find aberrant events. The authors divide the abnormal occurrences into two categories. The first was a local anomalous occurrence, when the behaviour of the locality differed from its spatiotemporal surroundings. The second was a global abnormal occurrence, where the entire scene was abnormal despite the fact that any given local behaviour could be normal. In order to propose a system to identify local and global events in congested environments, the authors use various types of bases. They suggested a brand-new feature description called Multi-scale histogram of optical flow to lay the groundwork for sparse representation. The more precise motion energy information is also provided by the multi-scale histogram of optical flow. They employed three different datasets to assess the effectiveness of their suggested algorithm. They employed the UMN dataset for the global abnormal event, and the UCSD Ped1 and Subway datasets for the local abnormal event. They use three measurements for evaluation: pixel, frame, and event levels. In comparison to other methods, the one suggested by the authors detects all wrong direction occurrences and also has a higher accuracy for no-payment events.

In [10] in the world of technology, the demand for intelligent video surveillance is rising since it can analyse huge amounts of video data. In order to uncover anomalies in crowded video sequences, the authors conducted research on abnormal event identification in crowded scenarios. They use a basic patch descriptor in their system, which divides a given check picture frame into two patches based on the size of the image frame and evaluates pixel motion energy. The projected system then used efficient edge detection and filtering DoG. The motion context was then allocated and normalised cuts and gaussian expectation-maximization techniques were employed to cluster related patches. In order to identify the anomalous activity scattered among the packed scenes, they ultimately adopted the k-nearest neighbour (k-NN) search method. In comparison to the current system, their proposed one takes less time to identify anomalies in a picture frame. The suggested system has a faster elapsed time and Ncuts computation time than the current system. The suggested system also enhances PSNR. Their technique efficiently and precisely aids in finding anomalies in video frames of busy scenes. However, due to several constraints, their technology is unable to process an entire video at once. Additionally, when patch size increases, system performance suffers.

In [11] the challenge of identifying unexpected patterns in recordings is known as "abnormal event detection" in surveillance. For quick abnormal event detection that was connected to subspace clustering, authors proposed an effective sparse combination learning framework using both batch and online solvers. They employed hierarchical clustering and k-means for subspace clustering. The framework they use to learn space basis combinations is described. Then they perform sparse combination training, where they evenly divide each layer into a collection of non-overlapping patches and resize each frame work into different scales to retrieve the relevant data. In terms of pixels, each patch is the same size. Birth-and-Death Combination Online Learning was the next stage. Then they do analysis for online training and testing. They create a novel evaluation dataset for their work. Their dataset contains a few plausible scenarios that were absent from earlier datasets. On the Subway dataset, they also perform quantitative comparisons with earlier approaches. For testing, a different UCSD Ped1 dataset is utilised. They employed the NYPD dataset along with the UMN data set to test the effectiveness of the online solver. They use a method that learns sparse combinations, which dramatically speeds up testing without sacrificing effectiveness. To manage massive amounts of data, an online solution was also made available.

In [12] processing a huge volume of surveillance videos required abnormal event detection. The task of spotting unusual events in crowded situations was difficult. Therefore, researchers conducted research on AED-NET, a network for the identification of aberrant events. They proposed a novel model to deal with the interconnectedness of human behaviour and to improve the representation of object's interaction in order to address how occlusion between moving objects affects the effectiveness when it was applied to crowded settings. They employed the deep learning network PCAnet, which can perform difficult tasks like facial recognition, for feature extraction. They chose this model because of its effectiveness and capacity for feature extraction competition. For a self-supervised learning technique for anomaly identification, they also employed kPCA. They computed optical flow using the Horn-Schunck (H-S) method. They made use of the numerous normal and aberrant occurrences found in the UMN dataset and the UCSD dataset. They constructed flow maps using the raw data from the surveillance video clips, and PCAnet then retrieved their high-level features, which were then utilised to assess the anomalism of both local and global anomalous events. The findings demonstrate how effectively the framework detects both local and global anomalous events.

In [1] it is stated that it has long been a difficult task to spot an anomaly in a crowded scene, but with the enormous growth of video there was an increasing need not only for object recognition and their behaviour, but in particular for detecting the rare, interesting occurrences of unusual objects or suspicious behaviour in the large body of ordinary data. Because of this, academics have done research on employing spatiotemporal auto-encoder to detect aberrant events in films. In their study, they employ a deep learning methodology to represent video data by a collection of standardised features that are taken from lengthy video clips. They used a domain-free approach that doesn't call for additional work from people. Their approach involved training an end-to-end model that combined a temporal encoder-decoder and a spatial feature extractor to understand the temporal patterns of the input volume of frames. First, they transform unaligned raw data into usable input for the model. To learn the regular patterns in the training films, they then suggest a convolutional spatio-temporal auto-encoder. Finally, after the model has been trained, the performance of the model may be assessed by feeding it test data to see if it was successful in recognising abnormal occurrences while maintaining a low false alarm rate. To train their model, they employed the UCSD D Ped1 and Ped2 datasets, the subway entrance and exit records, and the Avenue dataset.

In [13] it is stated that researchers in the field of computer science are paying more and more attention to video studies lately. There has been a lot of research towards the detection of abnormal events in crowded settings. The author's research on real-time abnormal event identification in crowded settings is presented in this publication. First, they developed the STACOG feature descriptor to characterise video events in order to recognise aberrant events in crowded settings. Second, they made use of the PETS2009 dataset and the UMN dataset, both of which are freely accessible surveillance datasets. Thirdly, their approach is computationally cheap, making it ideal for real-time anomaly identification in surveillance footage. The STACOG descriptor was employed for feature extraction. By taking advantage of the local correlations between the space-time gradients from the image sequences, it captures the geometric properties of a motion shape. They then divided the STACOG descriptors of the training frames into a set of clusters using the K-medoids clustering algorithm. On a laptop with an Intel Core i5-3210M CPU running at 2.50GHz and 6G RAM, they conducted trials to assess the precision and performance of the suggested approach. At the frame level, their suggested strategy has shown equivalent results, with 98.48 percent AUC for UMN and 92.13 percent accuracy for PETS 2009. However, they must research how to exploit the auto-correlation of streak flow features to enhance the proposed method's accuracy. By enclosing the velocity field for a while, streak flow faithfully captures the motions of the crowd.

## **Chapter 3**

# **Requirement Specifications**

This chapter deals with all the requirement specifications of the project, recording all the functional requirements and the non-functional requirements. A simple use-case has also been included at the end of this chapter, along with its dictionary.

### **3.1 Existing Systems**

Existing systems related to our project had somewhat similar features and working but some of the challenges in abnormal event detection include defining normal behaviors, addressing the variations in abnormal behavior, appropriate feature extraction, handling imbalanced distribution of normal and abnormal data, environmental variations, sparse occurrence of abnormal events and camera movements. Also we observed that facial identification is not on the top of the current object recognition as well as tracking for more enhanced surveillance capabilities.

Those systems were not able to overcome the problem of huge datasets to train the system perfectly and identifying anomalies other than motion anomalies seems difficult. Hence, we developed the system that will overcome some of the challenges like these.

### **3.2 Proposed System**

Our system automates the intimation of any such anomalies happened in a particular place like park, shopping mall and market etc. With the help of a video surveillance camera, which doesn't require any involvement of human beings to perform the intimation. This system then shall be propagated to other areas of identification like earthquakes. Our system classify events as normal or abnormal event through different classifiers like decision tree, naïve bayes, random forest, linear SVC and bagging.



### 3.3 Require Specification

Initially our system would be trained and tested for the selected set of abnormalities. Then the trained model would be integrated in such a way that would take input from the dataset and generate responses.

#### 3.3.1 Functional Requirements

Functional requirements are a set of requirements that describes how a system must function. It based on the proposed system and the users that are interacting with it. According to its elicited criteria, each system would have an own set of functional requirements. Our system has the following functional requirements.

##### 3.3.1.1 Import Video

Videos from dataset had been given in the form of input to the system to extract the frames of a specific threshold. The size of the extracted frames then is fixed as it helps in detecting the image.

##### 3.3.1.2 Pre-Processing

Before being used, data must first be processed. The idea of data preprocessing is to turn the raw data into a clean data set. Before submitting the dataset to the algorithm, it is pre-processed to look for missing values, noisy data, and other anomalies. So we have done pre-processing in this project in following steps:

- **Extract Frames** We extracted frames from both datasets. This process was done by using Video Capture Object. Avenue data set contains 15328 training, 15324 testing and 30652 frames in total. Web dataset contains 2545 abnormal, 5066 normal and 7611 frames in total. After extraction resize the frames of the videos to a fixed width and height of 128 and a sequence length of 16 ignoring the videos having frames less than the sequence length, to reduce the computations and normalize the resized frame by dividing it with 255 so that each pixel value then lies between 0 and 1. Later on these frames will used in detecting normal and abnormal events.
- **Filter Application** After extracting frames from videos, we applied filter for smoothing, modifying and enhancing the image. For filter application we used median filter. The median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. It replaces the pixel value with the median of neighboring pixel values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value.

- **Background Removal** Once the image is smoothed then we had removed the background. For background removal we used frame difference. Background subtraction as the name suggest is the process of separating out foreground objects from the background in a sequence of video frames. Firstly we converted the incoming frames to gray scale then subtract the current frame from the background model which is just the previous frame and last for each pixel calculate the difference between the current frame and the previous frame. If the difference between the current frame us greater then the threshold, the pixel is considered the part of foreground.

### 3.3.1.3 Detection of Objects

After the pre-processing phase, object detection is done using watershed algorithm. Watershed algorithm is based on extracting sure background and foreground and then using markers will make watershed run and detect the exact boundaries. This algorithm generally helps in detecting touching and overlapping objects in image. Firstly we had find the sure background using morphological operation like opening and dilation. Then we find the sure foreground using distance transform. and lastly unknown area is the area neither lies in foreground and background and used it as a marker for watershed algorithm.

### 3.3.1.4 Verification of Humans

Computer vision systems use human recognition and tracking to find and follow humans in video footage. Finding every occurrence of a human in an image is the challenge of human detection, which has most commonly been completed by scanning the entire image at all feasible scales and comparing a small region at each location with templates or patterns of individuals. In our approach, if the input video is correctly received, the system will accurately recognize the object using HOG.

### 3.3.1.5 Feature Extraction

Extra features of the dataset is extracted. Which includes:

- **Orientation** The system detects the orientation of the human. Meaning it determines the position in which the human is standing.
- **Movement** To estimate the motion pattern optical flow has been extracted for the selected pixels belonging to moving objects using Horn-Schunck optical flow algorithm. Speed of a pixels relates with its neighboring pixels. For every point in optical flow change of speed is smooth with no sudden changes.

- **Velocity** Distance of objects between two points has been computed using value of their centroid. Euclidean distance has been used to calculate distance. It is calculated by using the Euclidean distance formula given by equation. Position of pixels are used as initial and final stage of objects

#### **3.3.1.6 Model Training**

Model needs to be trained firstly. The system splits the data to create training and testing sets. It also shuffles the dataset before the split to avoid any bias and get splits representing the overall distribution of the data.

#### **3.3.1.7 Human Activity Recognition**

The goal of Human Action Recognition (HAR) is to comprehend human behavior and give each action a label. Due to its many uses, it has been garnering more and more interest in the computer vision community. According to our methodology, the system will identify each person's activity and designate it as either a normal or abnormal occurrence.

### **3.4 Non-Functional Requirements**

NFRs outline the constraints that have an impact on how the system should operate and fulfill the functional requirements. Here are a few NFRs connected to our system:

#### **3.4.1 Availability**

The system is available for anyone at every time.

#### **3.4.2 Reliability**

The system provides exact and authenticated data about the detection and recognition of video.

#### **3.4.3 Usability**

The system is available for usage when the user requests it. After choosing a video, the user checks to see what kind of event is taking place in it. The application then analyses the video and informs the user whether the occurrence is normal or abnormal.

#### **3.4.4 Security**

The system is also used for security purpose.

### 3.4.5 User Friendly:

The system is user friendly so that any person can use our system without any type of error.

## 3.5 Use Case Diagram

Use-cases are typically used to document the system requirements for a specific project, including the user experience and personnel involved. Figure demonstrates some of the functions of our system and how it communicates with users.

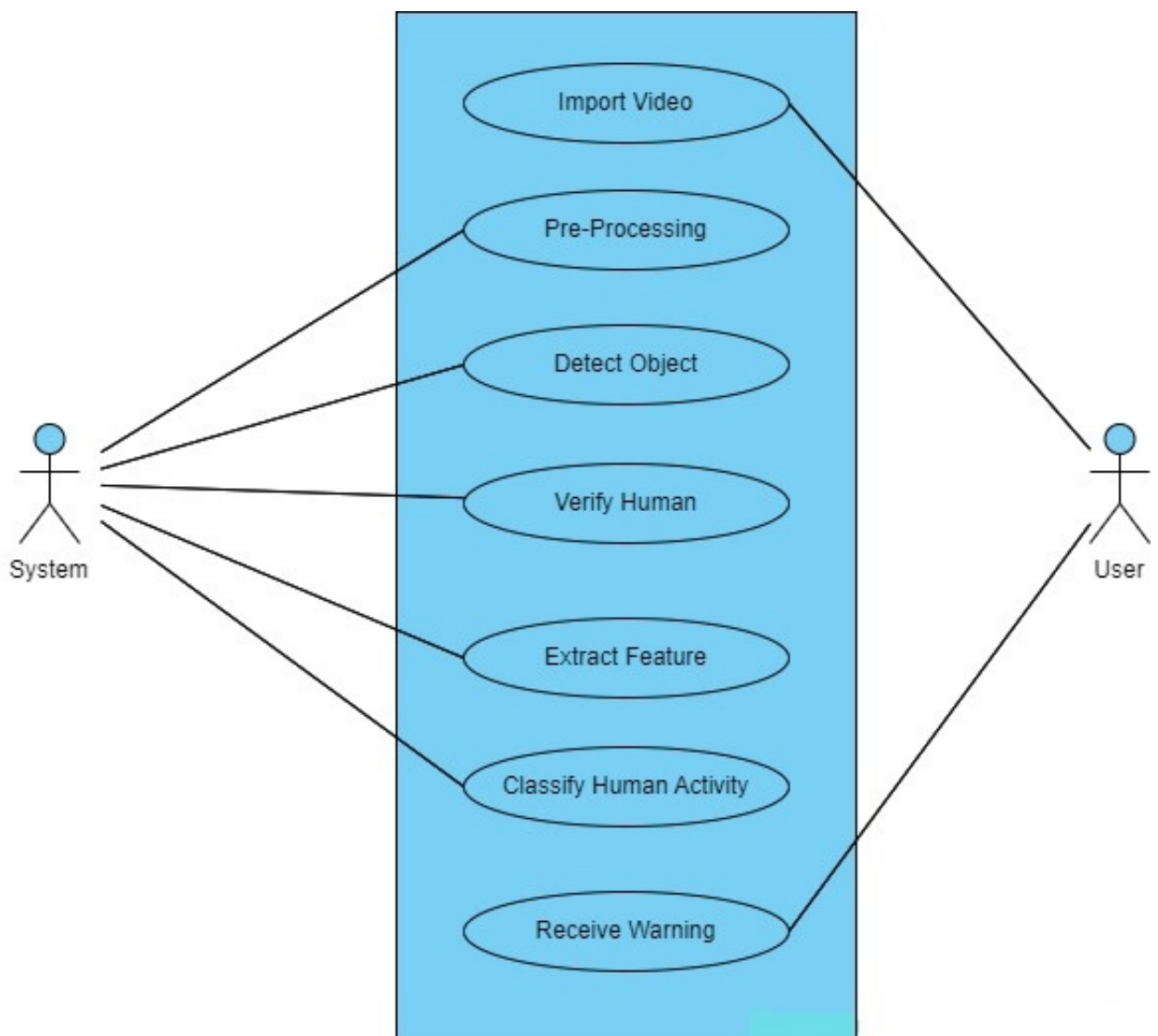


Figure 3.1: Use Case Diagram

### 3.5.1 Sub-Use Case Diagrams

The developed system accurately imports the videos stored in the local memory.



Figure 3.2: Use Case For Video Import

The system will display video as the video is uploaded.



Figure 3.3: Use Case For Video Display

When an anomaly is detected the system will generate an alert which is displayed on the front-end of web page.

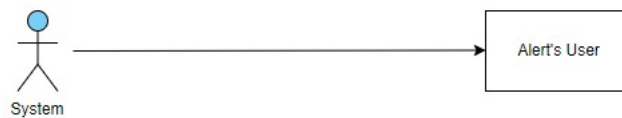


Figure 3.4: Use Case For Anomaly Alert

## 3.6 Use Case Dictionary

An explanation of each functionality in a use case diagram is provided by a use case dictionary. This greatly improves the reader's ability to comprehend the project's non-technical aspects.

### 3.6.1 Import Video

This is the initial version of the dictionary that concentrates on the section where videos are obtained that are utilized for anomaly identification.

Table 3.1: Use Case Dictionary for Import Video

Use Case ID	01
Title	Import Video
Pre-Condition	The system needs to have proper video clips that may be used for detecting the an anomaly.
Post-Condition	The video is imported successfully.
Basic Path	The user will upload the videos stored in the local memory for detecting the abnormal events.
Alternative Path	If videos are not uploaded successfully the user will try again.
Exceptional Path	Might be some errors or system failures.

### 3.6.2 Pre-Processing

In deep learning and data mining processes, it is essential that the project must first, pre-process our data to remove redundancies and unnecessary data. This makes our data more easy to read and more prune to a better output.

Table 3.2: Use Case Dictionary for Pre-Processing

Use Case ID	02
Title	Pre-Processing
Pre-Condition	We have chosen the proper dataset web and avenue to train the model with the video clips.
Post-Condition	The chosen dataset has video clips.
Basic Path	We import the video, read the content, and extract the frames from the video input and save them as images in the temporary memory for processing the same.
Alternative Path	Remove the background and noise for smoothing the image.
Exceptional Path	Might be some errors or system failures.

### 3.6.3 Object Detection

This portion of the project deals with the technical aspect. When the project has a video clip, it runs it through the model and detects all the objects present in the video and removes the unnecessary ones, including background noise, objects in the room like pictures.

Table 3.3: Use Case Dictionary for Object Detection

Use Case ID	03
Title	Object Detection
Pre-Condition	Valid and clear video is needed
Post-Condition	The objects are detected, and the noise and background objects are removed.
Basic Path	Extract the background and foreground of the image. For that purpose watershed was used.
Alternative Path	If the image is not pre-processed in case the image is big than the given threshold, then it is detected again.
Exceptional Path	Might be some errors or system failures.

### 3.6.4 Human Verification

The technological element of the project is covered in this section. When a video clip is available, the project will identify the object based on movement by pulling features from the frame.

Table 3.4: Use Case Dictionary for Human Verification

Use Case ID	04
Title	Verify Human
Pre-Condition	System is taking input video successfully.
Post-Condition	System will detect the Human by using HOG descriptor.
Basic Path	System will detect the object on the basis of the movements by extracting features from the frame.
Alternative Path	If the system will not detect the human on the basis of the movement of human then it will detect on the basis of velocity.
Exceptional Path	Frames are too large to extract the features of that object properly.

### 3.6.5 Feature Extraction

In this section system will extract the features precisely from the given dataset.



Table 3.5: Use Case Dictionary for Feature Extraction

Use Case ID	05
Title	Extract Features
Pre-Condition	System is detecting object successfully.
Post-Condition	System will extract the features precisely from given dataset.
Basic Path	System will extract features on basis of movement. Load the images normal and abnormal for further process.
Alternative Path	If system will not extract features on basis of above path then system will extract feature on basis of velocity of object.
Exceptional Path	Frames are too large to extract the features of that object properly.

### 3.6.6 Human Activity Recognition

In this section the system will recognize human activities based on their motion patterns in each frame.

Table 3.6: Use Case Dictionary for Human Activity Recognition

Use Case ID	06
Title	Recognize Human Activity
Pre-Condition	Identify the motion pattern of object in each frame.
Post-Condition	System will recognize the activity of every individual.
Basic Path	System will recognize the activity of human as normal event or abnormal event.
Alternative Path	For each input frame, we divide every scaled frame in equal and non-overlapping regions.
Exceptional Path	Scenario was extremely crowded to classify the event of individual object.

# **Chapter 4**

## **Design**

This chapter includes all the design for the said model, having an in-depth analysis of the needs and requirements. The chapter itself is divided into different modules, all of which are discussed in detail, below.

### **4.1 System Architecture**

This portion highlights the design at a low-level, including non-technical terms. This makes it easy to understand the general gist of the project, for those who are not familiar with the methods and concepts used, at root level. Our Project has no hardware requirements as its working is purely based on pre-defined datasets (pre-recorded videos). We used web and avenue dataset. We use Numpy, Pandas libraries along with OpenCV libraries. The model we proposed is further used for event classification.

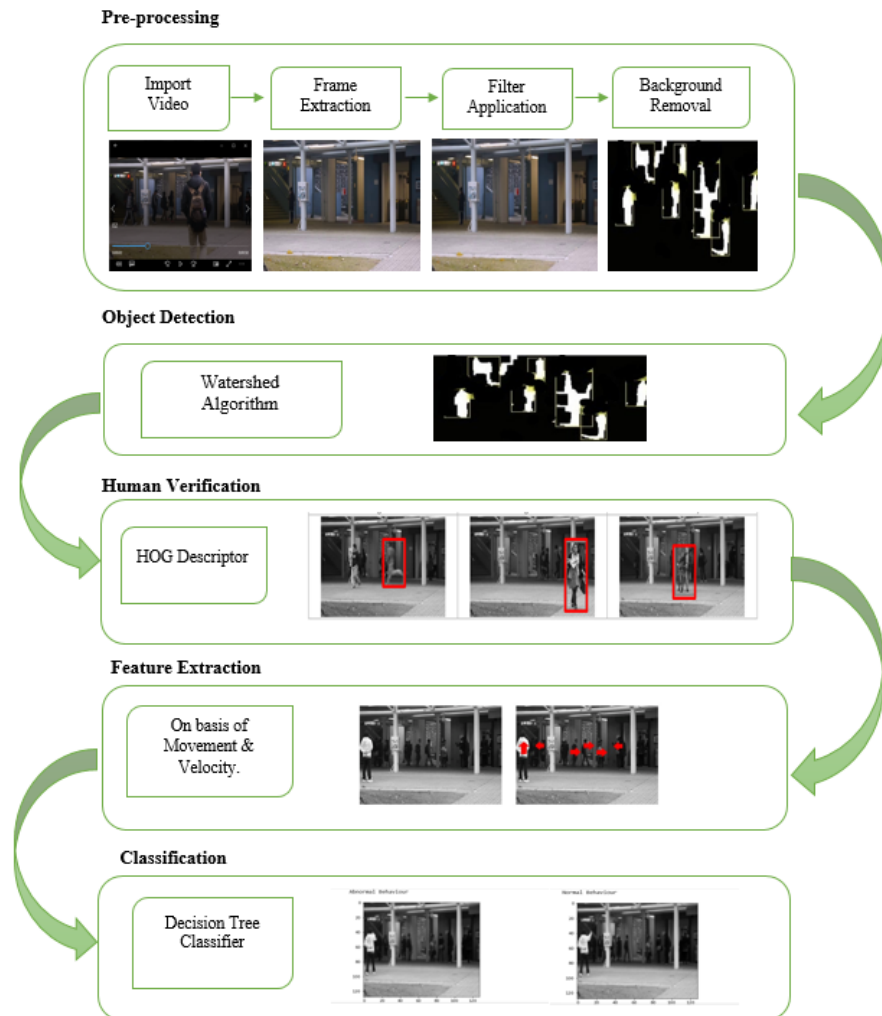


Figure 4.1: System Architecture

## 4.2 Design Constraints

There are a few constraints associated with the system.

- The browser needs to have access to the front-end website of the system that would act as a dashboard. Keeping this in mind the system should have access to internet for proper working.

## 4.3 Design Methodology

The objective of this system is to analyze videos, perform series of tasks for detection and recognition of actions from the set of predefined normal and abnormal situations. This system is developed on the basis of following activities.

- Import video from dataset.
- Frame extraction from the selected video.
- Apply median filter to smooth image.
- Remove background from the frame using frame difference method.
- Detect object in the frame using watershed algorithm.
- Verify human using HOG descriptor.
- Extract the features on the basis of movement and velocity to characterize various actions.
- Recognition of activity by using different classifier and results are shown to users.

To train the model we used proper dataset by providing video clips. In order to process the video, we import it, read the information, and then separate the frames from the video input and save them as images in the temporary memory. After extracting frames we had applied median filter to smooth the image and to reduce the noise present in the image. After that we removed the background using frame difference. Background subtraction as the name suggest is the process of separating out foreground objects from the background in a sequence of video frames. Then we used the HOG descriptor, which counts the instances of gradient orientation in focused areas of a picture, for human detection. After that on the basis of features of activity performed by human in video clips system recognized that activity is normal or abnormal event. For classification we used different classifiers linear svc, bagging, decision tree, naïve bayes and random forest.

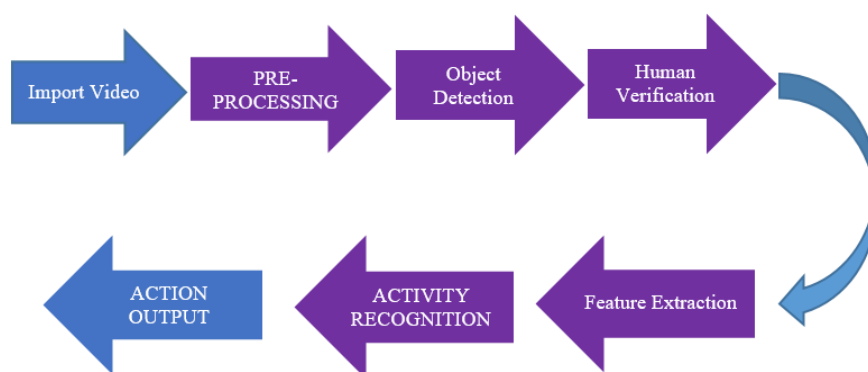


Figure 4.2: Methodology Steps

## 4.4 Sequence Diagram

A sequence diagram is a form of interaction diagram, because it illustrates the interactions between a group of items and the order in which they occur.

### 4.4.1 Human Detection

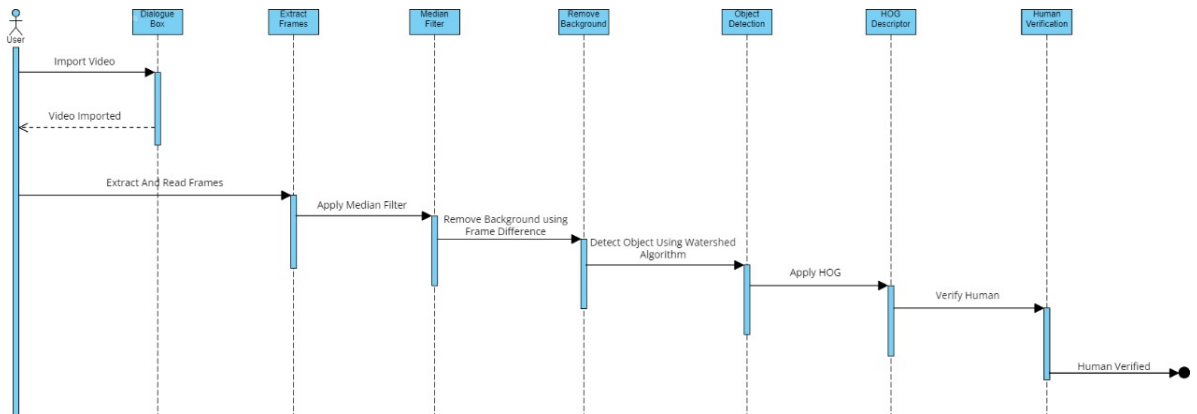


Figure 4.3: Human Detection

### 4.4.2 Activity Recognition

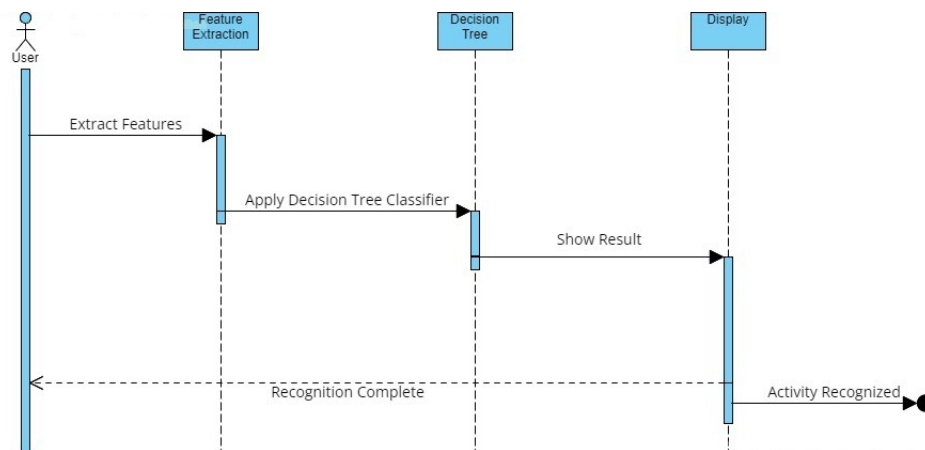


Figure 4.4: Human Activity Recognition

## 4.5 GUI Design

To make it easier for the user to run the script, this project needed an interactive user interface (UI). Making a Flask application that would serve as the front end and integrate it with the model and other Python scripts was the strategy used to complete this work. The project's UI is shown in figure below

### 4.5.1 User Interface 1



Figure 4.5: User Interface 1

### 4.5.2 User Interface 2



Figure 4.6: User Interface 2

### 4.5.3 User Interface 3

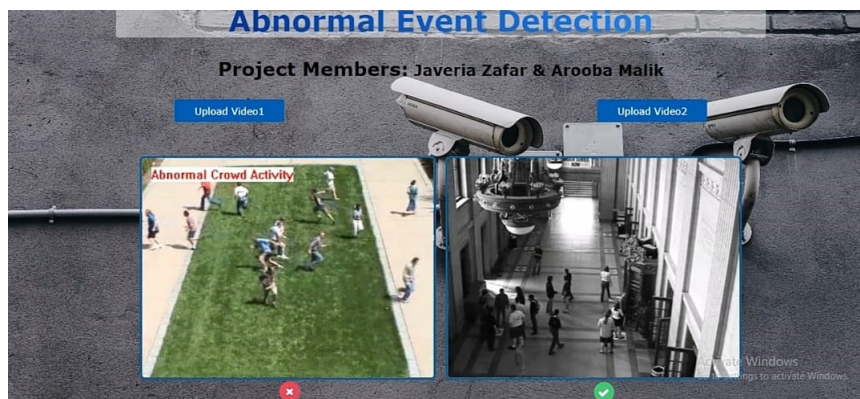


Figure 4.7: User Interface 3

### 4.5.4 User Interface 4



Figure 4.8: User Interface 4



## **Chapter 5**

# **System Implementation**

In this particular section of the thesis, it is diving in the physical implementation of the system and how it all works out as an end product. This is an essential portion as it covers the technical aspect of the project in depth.

### **5.1 Internal Component**

Anomaly detection has been a massive gap in the technical world and its introduction has opened up numerous gates for further research and development. Having said this, anomaly detection in videos, has been a tough ask since it is not possible to accurately depict what constitutes as an anomaly, given the wide nature. The system was trained in such a manner that it takes a video clip from the dataset having each frame extracted. The model then gives a frame score for each individual frame extracted from the video, by using Open-CV function, in the range of 0 to 1. if the value is 0, it indicates that they belong to abnormal behavior category and 1 shows it belong to normal behavior category. Our system is divided into 5 major components internally, which perform their task respectively. The first component is preprocessing in which the video is uploaded in order to pre-process by extracting frame from the video clip, nest step is object detection in which the object is detected, next our system will verify human. After that features are extracted and lastly classifiers are applied which gives out anomaly score.

## 5.2 Functionality of Components

### 5.2.1 Data Set

In this project we used two datasets i.e. WEB dataset and Avenue dataset. Both contain normal and abnormal video clips. We use 80% dataset for training purpose and 20% for testing the model.

Avenue Dataset contains 21 testing and 16 training video clips. The videos are captured in CUHK campus avenue with 30652 (15328 training, 15324 testing) frames in total.



Figure 5.1: Avenue Dataset

The web dataset contains 8 Videos of real-life escape panic, clash, fight and 12 Videos of normal pedestrians. This dataset contains 2545 abnormal, 5066 normal and 7611 frames in total.



Figure 5.2: Web Dataset

## 5.2.2 Pre-processing

Before being used, data must first be processed. To pre-process the data the system read the video files from the dataset and resize the frames to a fix width and height of 128 and a sequence length of 16 while disregarding the videos having frames shorter than the sequence length to reduce computations. We then normalize the resized frame by dividing it with 255 so that each pixel value then lies between 0 and 1. The pre-processing in this project includes following steps:

### 5.2.2.1 Import Video

Since this project revolves around existing system, the project needs to have an input source that can give us video clip or video input. Figure 5.1 shows an example of a locally stored video.

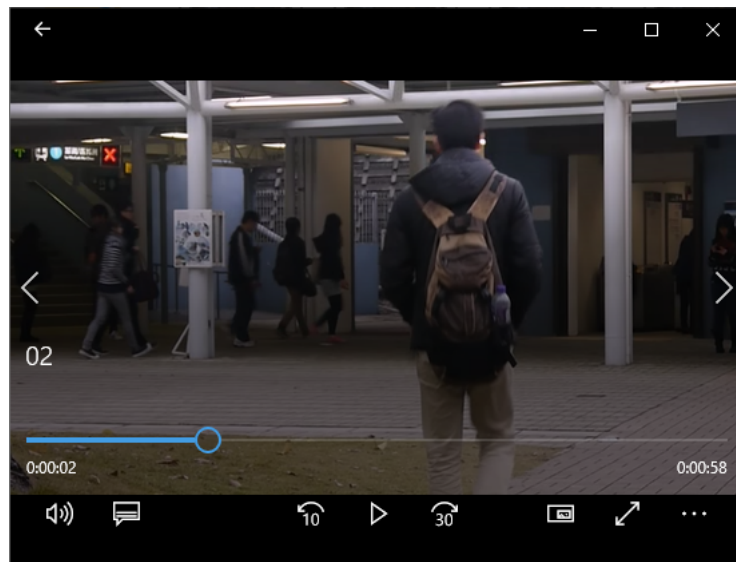


Figure 5.3: Video Input

### 5.2.2.2 Extract and Read Frames

Now we extract frames from both datasets. The process was done by using Video Capture Object. Avenue data set contains 15328 training, 15324 testing and 30652 frames in total. Web dataset contains 2545 abnormal, 5066 normal and 7611 frames in total. Later On these images will used in detecting normal and abnormal events.

### 5.2.2.3 Filter Application

After extracting frames from videos, we applied filter for smoothing, modifying and enhancing the image. For filter application we used median filter. The median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. It replaces the pixel value with the median of neighboring pixel values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value.

### 5.2.2.4 Remove Background

Once the image is smoothed then we had removed the background. For background removal we used frame difference. Background subtraction as the name suggest is the process of separating out foreground objects from the background in a sequence of video frames. Firstly we converted the incoming frames to gray scale then subtract the current frame from the background model which is just the previous frame and last for each pixel calculate the difference between the current frame and the previous frame. If the difference between the current frame us greater then the threshold, the pixel is considered the part of foreground.

### 5.2.3 Object Detection

After the pre-processing phase, object detection is done using watershed algorithm. Watershed algorithm is based on extracting sure background and foreground and then using markers will make watershed run and detect the exact boundaries. This algorithm generally helps in detecting touching and overlapping objects in image. Firstly we had find the sure background using morphological operation like opening and dilation. Then we find the sure foreground using distance transform. and lastly unknown area is the area neither lies in foreground and background and used it as a marker for watershed algorithm.

### 5.2.4 Human Verification

Human Verification is the task of locating all instances of human beings present in an image, and it has been most widely accomplished by searching all locations in the image,

at all possible scales, and comparing a small area at each location with known templates or patterns of people.

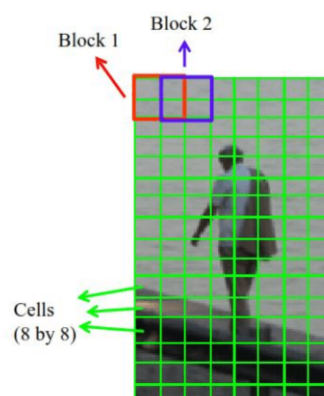
#### 5.2.4.1 HOG

The Histogram of Oriented Gradients (HOG) is a feature descriptor used in computer vision and image processing applications for the goal of the object detection. It is a method that counts events of gradient orientation in a specific portion of an image or region of interest.

In 2005, Dalal and Triggs published a research paper named Histograms of Oriented Gradients for Human Detection. After the release of this paper, HOG is used in a lot of object detection applications.

Here are the most important aspects of HOG:

- HOG focuses on the structure of the object. It extracts the information of the edges magnitude as well as the orientation of the edges.
- It uses a detection window of 64x128 pixels, so the image is first converted into (64, 128) shape.
- The image is then further divided into small parts, and then the gradient and orientation of each part is calculated. It is divided into 8x16 cells into blocks with 50 percent overlap, so there are going to be  $7 \times 15 = 105$  blocks in total, and each block consists of 2x2 cells with 8x8 pixels.
- We take the 64 gradient vectors of each block (8x8 pixel cell) and put them into a 9-bin histogram.



#### 5.2.4.2 Initializing the HOG Person

Firstly, initialize the HOG (Histogram of oriented gradient) person. Object detection frequently makes use of the HOG features. An image is divided into small squared cells

using HOG, which then computes a histogram of oriented gradients in each cell, normalises the outcome using a block-wise pattern, and provides a description for each cell. HOG is often used for object detection. Take the input image you want to calculate HOG features of. Resize the image into an image. Detect all regions in the image that has a person inside it. Draw original bounding boxes around the object. Now show the output.



Figure 5.4: Histogram of oriented gradient

### 5.2.5 Feature Extraction

Feature extraction is a step of the dimensional reduction process which divides and condenses an initial set of raw data into smaller, easier-to-manage groups. When you have a large data set and need to conserve resources without losing any important or significant information, the feature extraction technique can be helpful. The amount of redundant data in the data collection is decreased with the help of feature extraction.

In the end, the reduction of the data speeds up the learning and generalisation phases of the machine learning process and makes it easier for the computer to develop the model.

In this section, we discuss the features we used to classify the events. We have extracted features on the basis of velocity and movement.



Figure 5.5: Feature Extraction

#### 5.2.5.1 Velocity

Distance of objects between two points has been computed using value of their centroid. Euclidean distance has been used to calculate distance. It is calculated by using the Euclidean distance formula given by equation. Position of pixels are used as initial and final stage of objects.

where  $x_1$  is position of previous pixel,  $x_2$  is position of present pixel in terms of width,  $y_1$  is the position of previous pixel and  $y_2$  is position present pixel in terms of height. Considering the distance velocity of all moving objects have been calculated using distance travelled per unit time with respect to frame rate as equation below. The velocity of the object is presented .

### 5.2.5.2 Movement

To estimate the motion pattern optical flow has been extracted for the selected pixels belonging to moving objects using Horn-Schunck optical flow algorithm. Speed of a pixels relates with its neighboring pixels. For every point in optical flow change of speed is smooth with no sudden changes. Smoothing constraint has been described. Figure presents the optical flow of multiple points selected.



### 5.2.6 Classification

The human activity categorization problem has remained a challenging task in computer vision for more than two decades. Previous works on characterizing human behavior have shown great potential in this area.

For classification of event whether it's normal or abnormal. We use five different classifiers decision tree, Naïve Bayes, Random forest, linear SVC and OVR bagging.

```
ovr_decision_tree_classifier_object.fit(training_data_features,training_output_only)
model_predictions = ovr_decision_tree_classifier_object.predict(testing_data_features)
lg_accuracy_scores = accuracy_score(testing_output_only, model_predictions)
```

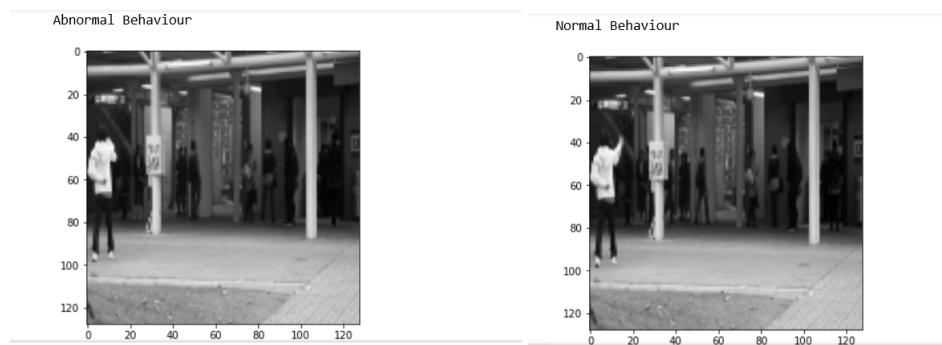


Figure 5.6: Classification

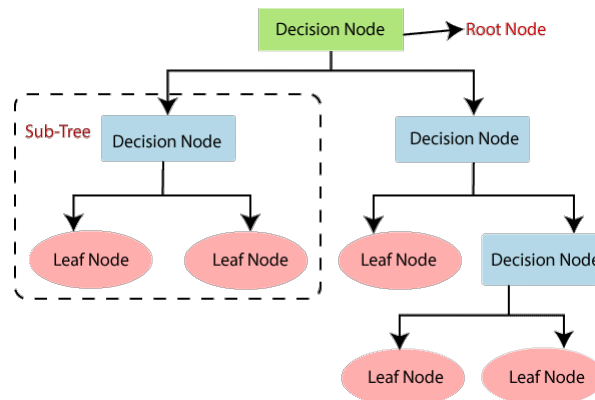
### 5.2.6.1 Decision Tree

- **Definition** A helpful supervised machine learning tool, Decision Tree can be applied to both classification and regression issues. They are effective analytical models that require little pre-processing time to understand data. It is a decision-support tool with a tree-like structure that offers suggestions for potential outcomes and costs of choices.
- **Construction** A decision tree displays a decision-making algorithm in a structure analogous to a flowchart. By separating the source set into subgroups based on a quality worth test, a tree can be comprehended. Recursive partitioning is the process of repeating this cycle on each selected subset. When the value of the subset at the node matches the value of the subset of the target variable or when splitting does not improve the accuracy of the predictions, the recursion is said to be complete. The creation of a decision tree classifier does not require any expertise in boundary setting or the domain. High dimensional data can be handled via decision trees. Overall, the decision tree classifier has excellent precision and uses an inductive method to learn about characterization.
- **Representation** Decision trees classify or organise events by placing them from the tree's root to various leaf nodes. This offers a description or classification of the occurrences. An occurrence is categorised by starting at the root node of the tree, examining the characteristics suggested by this node, and then moving along the tree branch by evaluating it against the value of the attribute. For the subtree that is rooted at the new node, the procedure is repeated.



- **Example** Here is an illustration of a binary tree model. Suppose you want to know if a person is fit based on the information they have provided about their age, food habits, physical activities, etc. The questions "what's the age," "does he work out?" and "does he eat too many pizzas?" would be the decision nodes in this case. The leaves, which are the actual outcomes, are also either "fit" or "unfit." It is classified as binary because there are just two options.
- **Advantages** Decision trees have a number of benefits over other decision-making tools. Among them are:
  - It is not necessary to standardise or normalise the acquired data before using a decision tree method. Both continuous and categorical variables are supported.
  - It is not necessary to credit the missing data when using the decision tree algorithm.
  - Pre-processing procedures in a decision tree making model demand less code and analysis than typical pre-processing steps of data.
  - In contrast to the typical data pre-processing processes, the pre-processing phases in a decision tree making model save time.
  - A decision tree generates comprehensive rules.
  - Compared to other algorithms, the principle or concept that underpins the decision tree making model is simpler and more recognisable to developers and programmers.
- **Disadvantages**
  - A decision tree typically requires more memory because of the numerical calculations involved.
  - A decision tree's repeatability is extremely sensitive since even a small change in the data can have a significant impact on the tree's structure.
  - Decision tree making methods are less suitable for jobs requiring the prediction of continuous attribute values due to their limitations.
  - Making a decision tree is highly expensive because fields need to be sorted at each node. Some algorithms combine the usage of several distinct fields at once, thus raising expenses.

- **Conclusion** It is evident from the description above that decision trees can effectively handle non-linear data sets. In many spheres of life, including engineering, civil planning, business, and even law, it acts as a stimulant for decision-making. It is necessary to consider the decision tree model's benefits and drawbacks in light of how well they fit the current problem statement.



#### 5.2.6.2 Naïve Bayes

- **Definition** It is a classification method built on the Bayes Theorem and predicated on the idea of predictor independence. A Naive Bayes classifier, to put it simply, believes that the presence of one feature in a class has nothing to do with the presence of any other feature. Simple to construct and especially helpful for very big data sets is the naive Bayes model. Along with being straightforward, Naive Bayes is known to perform better than even the most complex classification techniques.
- **Example** A fruit might be categorised as an apple, for instance, if it is red, rounded, and around 3 inches in diameter. Even if these characteristics depend on one another or on the presence of other characteristics, each of these traits separately increases the likelihood that this fruit is an apple, which is why it is called "Naive."
- **Pros**
  - The test data set class can be predicted quickly and easily. Additionally, it excels at multi-class prediction.
  - A Naive Bayes classifier performs better than other models like logistic regression when the assumption of independence is true, and it requires fewer training data.

- Compared to a numerical variable, it performs well with categorical input variables (s). It is assumed that numerical variables have a normal distribution (bell curve, which is a strong assumption).

- **Cons**

- If a categorical variable has a category in the test data set but not in the training data set, the model will give it a probability of 0 (zero) and will not be able to predict anything. This is commonly referred to as "Zero Frequency." We can utilise the smoothing method to resolve this. Laplace estimate is one of the simplest smoothing methods.
- However, naive Bayes is likewise regarded as a poor estimator, therefore the predict proba outputs for probability should not be taken too seriously.
- The assumption of independent predictors is one more Naive Bayes flaw. We rarely find a set of predictors that are entirely independent in the real world.

- **Applications of Naive Bayes Algorithms**

- Real time Prediction:  
Naive Bayes is a quick classifier that eagerly learns new things. As a result, it might be applied to real-time prediction.
- Multi class Prediction:  
The ability of this method to predict many classes is very widely recognised. Here, we can forecast the likelihood of several target variable classes.
- Text classification/ Spam Filtering/ Sentiment Analysis:  
Because they perform better in multi-class situations and follow the independence criterion, naive Bayes classifiers are frequently employed in text classification and have a greater success rate than other methods. It is therefore frequently used in Sentiment Analysis and Spam Filtering (to identify spam e-mail) (in social media analysis, to identify positive and negative customer sentiments)
- Recommendation System:  
Together, Naive Bayes Classifier and Collaborative Filtering create a recommendation system that filters opportunistic information and forecasts whether a user would find a given resource appealing or not.

### 5.2.6.3 Random Forest

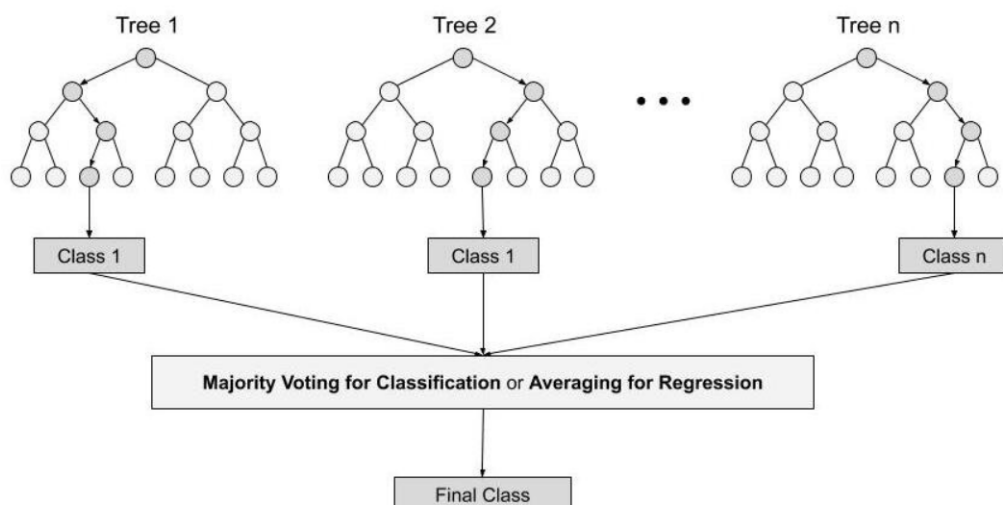
- **Introduction** Supervised machine learning algorithms like random forest are frequently employed in classification and regression issues. On various samples, it constructs decision trees and uses their average for classification and majority vote for regression.

The Random Forest Algorithm's ability to handle data sets with both continuous variables, as in regression, and categorical variables, as in classification, is one of its most crucial qualities. In terms of classification issues, it delivers superior outcomes.

- **Real Life Analogy** To better comprehend this idea, let's use a real-world analogy. When it comes to selecting a course based on his skill set, a student named X is confused. So he makes the decision to talk to a variety of people, including his cousins, teachers, parents, degree students, and workers. He asks them a range of queries, such as why he ought to select, employment prospects with that course, tuition, etc. He ultimately chooses to take the course that the majority of people recommended after consulting with a variety of people about it.

- **Steps involved in random forest algorithm**

- Step 1: In Random Forest, n records at random are selected from a data set with k records.
- Step 2: Each sample's decision tree is built separately.
- Step 3: The output of each decision tree will be produced.
- Step 4: For classification and regression, the final result is based on the majority vote or average, respectively.



- **Features**

- Diversity: Each tree is unique, so not all qualities, variables, or features are taken into account while creating a particular tree.
- Immune to the curse of dimensionality: The feature space is smaller since no tree takes into account all features.
- Train-Test split: In a random forest, there will always be 30 percent of the data that the decision tree cannot see, therefore we don't need to separate the data for train and test.
- Stability: Stability arises because the result is based on majority voting/ averaging.

- **Advantages**

- Because the result is based on majority vote or average, it solves the over-fitting problem.
- Even if the data has null or missing values, it still works properly.
- Because each decision tree that is produced is independent of the others, it demonstrates the parallelization property.

- **Disadvantages**

- When compared to decision trees, where choices may be made by going down the tree's path, random forests are much more complex.
- As a result of its complexity, training takes longer than for other models. Every time a decision tree needs to make a prediction, it must produce output based on the input data.

- **Conclusion** We may now draw the conclusion that Random Forest is one of the greatest, high-performing strategies that is widely employed in many industries due to its effectiveness. It can handle category, continuous, and binary data. If someone wants to create a model quickly and effectively, random forest is an excellent option because one of its strongest features is that it can manage missing values.

Overall, random forest is a quick, straightforward, adaptable, and reliable model with certain drawbacks.

#### 5.2.6.4 Linear SVC

- **Introduction** The supervised machine learning technique known as SVC, or Support Vector Classifier, is frequently used for classification problems. A technique called the Linear Support Vector Machine (Linear SVC) looks for a hyperplane to maximise the distance between samples that are classified.

With a high number of data, the Linear Support Vector Classifier (SVC) approach performs well. It uses a linear kernel function to perform classification. When compared to the SVC model, the Linear SVC adds more parameters such as penalty normalisation, which uses the loss function. Because linear SVC is based on the kernel linear technique, the kernel method cannot be modified.

- **Objective** A Linear SVC's (Support Vector Classifier) goal is to split or categorise the data you supply by returning a "best fit" hyperplane. You may then feed some features to your classifier to get the "predicted" class after acquiring the hyperplane. This makes this particular algorithm—which may be used in a variety of circumstances—pretty ideal for our purposes.
- **Working** SVC separates the data into two classes by mapping the data points to a high-dimensional space and then locating the best hyperplane. Scikit-version learns of SVC, known as Sklearn SVC, is a well-known machine learning library.
- **Margin** The criterion is SVC if the hyperplane we are utilising for classification is in linear condition.

The margin, which is a line's separation from the nearest class points, is the separation of the vectors from the hyperplane. The margin between classes should be maximised in the hyperplane we select. Margin can be further classified into:

- **Soft Margin:** We will let some margin violation to occur, which is known as soft margin classification, because the majority of real-world data are not totally linearly separable.
- **Hard Margin:** If the training data can be separated into two classes linearly, we can choose two parallel hyperplanes to divide the classes, increasing the distance between them.

### 5.2.6.5 Bagging

- **Introduction** An ensemble learning technique called bagging, often referred to as Bootstrap aggregating, aids in enhancing the efficiency and precision of machine learning algorithms. It lowers the variance of a prediction model and is used to handle bias-variance trade-offs. Bagging, specifically decision tree methods, is used for both regression and classification models to prevent over-fitting of the data.

Bootstrapping is a technique for estimating a population parameter by randomly selecting data samples from a population.

- **Steps to Perform Bagging**

- Consider a training set that contains  $m$  features and  $n$  observations. You must choose a random sample without replacement from the practise dataset.
- Using sample data, a model is constructed using a subset of  $m$  features that is randomly selected.
- The nodes are divided using the feature that offers the best split among all of them.
- You have the best root nodes because the tree has matured.
- Repeating the previous steps  $n$  times. To provide the most accurate prediction, it combines the results of various decision trees.

- **Advantages**

- Bagging reduces data over-fitting.
- It raises the model's precision.
- It effectively handles higher dimensional data.

- **Conclusion** In statistics and machine learning, the notion of bagging is significant because it prevents data from becoming overfit. It is a model averaging technique that can be used with other algorithms in addition to decision trees.

### 5.2.7 Accuracy Generated From Model

Classifiers	Accuracy
Bagging	0.96
Decision Tree	0.96
Linear SVC	0.93
NB	0.93
Random Forest	0.96

Table 5.1: Accuracy Results

Unnamed: 0	Accuracy	Precision	Recall	Weighted-F1	Micro-F1	Macro-F1
0 OVR-Bagging	0.96	0.95	0.97	0.96	0.95	0.96
1 OVR-Decision-Tree	0.96	0.95	0.97	0.96	0.95	0.96
2 OVR-Linear-SVC	0.93	0.91	0.92	0.93	0.91	0.92
3 OVR-NB	0.93	0.91	0.93	0.93	0.91	0.92
4 OVR-Random-Forest	0.96	0.95	0.97	0.96	0.95	0.96

Figure 5.7: Accuracy

## 5.3 Libraries

### 5.3.1 OS

The OS module in Python has functions for adding and deleting folders, retrieving their contents, changing the directory, locating the current directory, and more.

Before you can communicate with the underlying operating system, you must import the os module. In order to use its functions, import it first using the import os command.

### 5.3.2 Numpy

An open-source library for the Python programming language is called NumPy (Numerical Python). It is employed in array manipulation and scientific computing. In addition to a multidimensional array object, it offers high-level array manipulation features.

Numerous mathematical operations can be carried out on arrays with NumPy. It provides a vast library of high-level mathematical functions that work on these arrays and matrices, as well as strong data structures that ensure efficient calculations with arrays and matrices.



### 5.3.3 Pandas

The most often used open source Python library for data science, data analysis, and machine learning activities is called Pandas. It is constructed on top of Numpy, a different package that supports multi-dimensional arrays. It is a data analysis and manipulation software package created for the Python programming language. It includes specific data structures and procedures for working with time series and mathematical tables.

### 5.3.4 CV2

An excellent tool for image processing and computer vision work is OpenCV. It is a free library that may be used to carry out operations like face recognition, object tracking, landmark recognition, and many other things. It supports many languages, such as Python, Java, and C++.

A collection of Python bindings called OpenCV-Python was created to address issues with computer vision. The `imread()` function of CV2. loads an image from the given file. This method produces an empty matrix if the picture cannot be read (due to a missing file, poor permissions, an unsupported or invalid format, etc.).

### 5.3.5 Glob

The Python glob module allows us to search over all path names in order to find files that fit a given pattern (which is defined by us). The rules established by the Unix shell are used to define the supplied pattern for file matching. The output of the software returns the result acquired by adhering to these guidelines for a certain pattern file matching in the random order. Because the glob module can navigate through the list of files at a specific point in our local disc, we must meet certain requirements when using the file matching pattern. The module will mostly review those lists of files on the disc that have a particular pattern.

### 5.3.6 Scikit-learn

In the Python ecosystem, Scikit-learn, an open source data analysis toolkit, is considered to be the pinnacle of machine learning (ML). Important ideas and traits include: algorithms for making decisions, such as: Data are identified and categorised by classification based on patterns. The most effective and reliable Python machine learning library is called Sklearn (Skit-Learn). Through a Python consistency interface, it offers a variety of effective tools for statistical modelling and machine learning, including classification, regression, clustering, and dimensionality reduction.

## **Chapter 6**

# **System Testing and Evaluation**

In this portion of the document, it will look at the evaluation criteria and testing of the whole integrated system. Testing is a crucial phase of any project development and in order for any project to be deployed, it needs to go through extensive testing.

### **6.1 Graphical User Interface Testing**

Figure shows what the GUI looks like after completion. The system is made using simple HTML and CSS language integrated with the trained Decision Tree model. There are a few components associated with the whole system and tests each of them individually.

#### **6.1.1 Video Input**

The first phase was to test if the GUI was able to load a video from the system and read it in the model. It has an option at the top of the web page that asks the user to choose a video file from the computer and displays the title of that video. This test was conducted successfully.

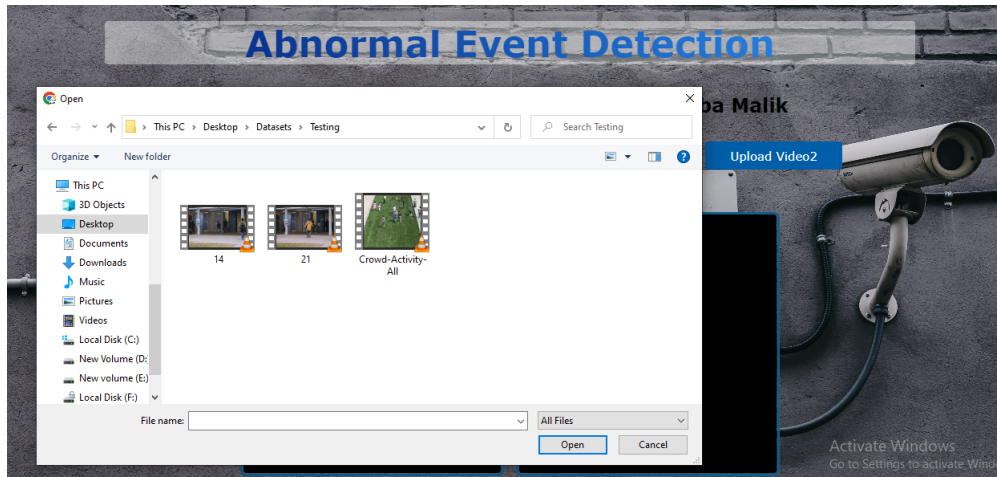


Figure 6.1: Load Image

### 6.1.2 Video Display

Next it had to display the selected video to the front-end. Any video input that the user passed in the website, if its locally stored then it can display it and the user can counter-check to see if that was indeed what he selected.

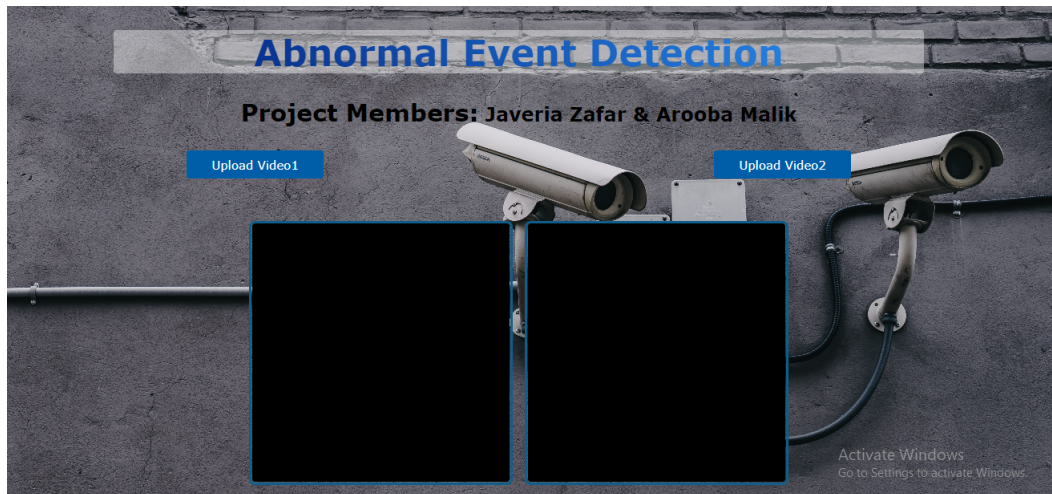


Figure 6.2: Without Video



Figure 6.3: With Video

### 6.1.3 Anomaly Detection Alert

A heading tag was created which would display alerts when an anomaly is being detected in the video stream. The python script was given a threshold of 0 which means that any frame with an anomaly score that was 0, is to be considered as an anomalous frame and hence an alert should be given out.



Figure 6.4: Detection Alert





Figure 6.5: Detection Alert

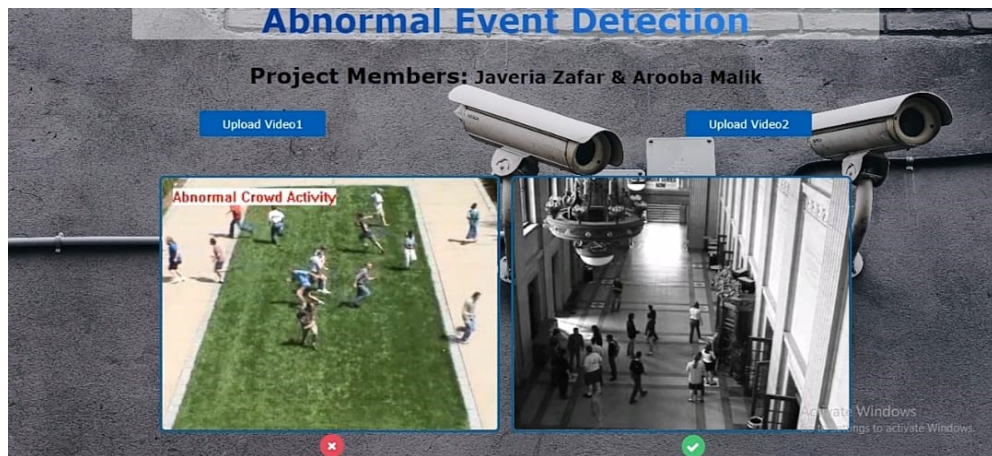


Figure 6.6: Detection Alert



Figure 6.7: Detection Alert

## 6.2 Usability Testing

In this phase of testing it deduced how easy it was to perform the desired tasks by using the project's web interface. It included simple buttons which perform their tasks on a single click and nothing complicated was included in the front end.

## 6.3 Test Cases

Multiple test cases were carried out on the interface and model, and the results were noted down in a tabular form. The following tables give a more detailed and descriptive analysis of the test cases.

### 6.3.1 Model Training

The decision tree model was first tested using the provided Avenue and Web dataset. The table below displays the outcomes.

Table 6.1: Test Case for Train Model

Test Case ID	01
Title	Train Model
Initial State	Avenue and Web dataset had to be available
Input	Avenue and Web dataset
Expected Output	Model should be successfully trained on data
Actual Output	Model was successfully trained on data
Status	Pass

### 6.3.2 Model Testing

Next it had to test the trained model and validate the results. The results are shown in the table below.

Table 6.2: Test Case for Test Model

Test Case ID	02
Title	Test Model
Initial State	Trained model had to be available
Input	Video Input
Expected Output	Model gives categorical accuracy
Actual Output	Model gave categorical and validation accuracy of 0.96
Status	Pass

### 6.3.3 Video Input

In the interface, a button was created that would allow the user to take video input from his local device. The table below displays the outcomes.

Table 6.3: Test Case for Video Input

Test Case ID	03
Title	Video Input
Initial State	No video
Input	Local video input
Expected Output	Video should be displayed on interface
Actual Output	Title of video was displayed
Status	Pass

#### 6.3.4 Load Multiple Video

The interface was checked if it had the ability to take and load multiple video input. The table below displays the outcomes.

Table 6.4: Test Case for Load Multiple Videos

Test Case ID	04
Title	Load Multiple Videos
Initial State	No video loaded
Input	Avenue and Web dataset
Expected Output	Load multiple videos
Actual Output	Interface could load multiple videos
Status	Pass

#### 6.3.5 Frame Extraction

The next step was to use OpenCv library to extract frames from a video input. The table below displays the outcomes.



Table 6.5: Test Case for Frame Extraction

Test Case ID	05
Title	Extract Frames
Initial State	Video input had to be available
Input	Video input
Expected Output	Frames should be extracted
Actual Output	Frame extraction was successful
Status	Pass

### 6.3.6 Processing

The interface has a headings tag that displays alerts in real time. The results are shown in the table below.

Table 6.6: Test Case for Processing

Test Case ID	06
Title	Processing
Initial State	Heading is not visible
Input	Frame processing
Expected Output	Output of each frame in real-time
Actual Output	Alerts are visible on the web page
Status	Pass

### 6.3.7 Video Display

In order to make the process synchronous, the project displayed video as well on the web page, in real-time. The table below displays the outcomes.

Table 6.7: Test Case for Video Display

Test Case ID	07
Title	Video Display
Initial State	No video displayed
Input	Video input
Expected Output	Video displayed
Actual Output	Video was displayed successfully
Status	Pass

### 6.3.8 Anomaly Alert

The project includes a heading tag in the web page that shows us if the video input has an anomaly or not. The results are shown in the table below.

Table 6.8: Test Case for Anomaly Alert

Test Case ID	08
Title	Anomaly Alert
Initial State	No alert is given
Input	Anomaly score and threshold
Expected Output	Alert should be successfully generated
Actual Output	Alert was successfully generated
Status	Pass

## 6.4 Exceptional Handling

There are also some limitations due to chosen of web dataset, as our target was small to medium density crowd. Web dataset was for high density. If any one want to made this system again, then they choose according to their target like small or High density crowd.

## Chapter 7

# Conclusions

One of the most important tasks in research applications is abnormal event detection. There has to be further research into discriminating techniques or motion information because both normal and aberrant occurrences share certain similarities. We suggested a model that may be used for event detection for that reason. The system will determine if an occurrence is normal or abnormal based on how people move. HOG descriptor is used for human detection. We employ five distinct classifiers: decision tree, naive bayes, bagging, linear SVC, and random forest for classification. Avenue and the Web dataset are the two datasets we utilise to assess the performance of our approach. Our system has an accuracy of 0.96. In order for our model to be accurate and demonstrate improvement

The project assisted us in enhancing our abilities in a variety of areas, including Python programming, artificial intelligence, and traditional and machine learning based. As we worked on this project, we gained a lot of knowledge. We learned how the machine learning algorithms function and how to simply modify them through training and testing the model. We gained insight into the field of computer vision by using the Open-Cv package. Additionally, we were able to gain a lot of knowledge on how to combine Python with HTML, CSS, and JS.

# References

- [1] Yong Shean Chong and Yong Haur Tay. Abnormal event detection in videos using spatiotemporal autoencoder. pages 189–196, 2017. Cited on pp. 1 and 11.
- [2] Kang Hao Cheong, Sandra Poeschmann, Joel Weijia Lai, Jin Ming Koh, U Rajendra Acharya, Simon Ching Man Yu, and Kenneth Jian Wei Tang. Practical automated video analytics for crowd monitoring and counting. *IEEE Access*, 7:183252–183261, 2019. Cited on p. 1.
- [3] Kothapalli Vignesh, Gaurav Yadav, and Amit Sethi. Abnormal event detection on bmtt-pets 2017 surveillance challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 36–43, 2017. Cited on p. 2.
- [4] Thittaporn Ganokratanaa, Supavadee Aramvith, and Nicu Sebe. Unsupervised anomaly detection and localization based on deep spatiotemporal translation network. *IEEE Access*, 8:50312–50329, 2020. Cited on p. 2.
- [5] Tahjid Ashfaque Mostafa, Jia Uddin, and Md Haider Ali. Abnormal event detection in crowded scenarios. In *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–6. IEEE, 2017. Cited on p. 7.
- [6] Shiyang Yan, Jeremy S Smith, Wenjin Lu, and Bailing Zhang. Abnormal event detection from videos using a two-stream recurrent variational autoencoder. *IEEE Transactions on Cognitive and Developmental Systems*, 12(1):30–42, 2018. Cited on p. 8.
- [7] Mona Izadi, Zohreh Azimifar, and Gholam-Hossein Jowkar. Abnormal event detection in indoor video using feature coding. In *2017 Artificial Intelligence and Signal Processing Conference (AISP)*, pages 151–155. IEEE, 2017. Cited on p. 8.
- [8] Sangmin Lee, Hak Gu Kim, and Yong Man Ro. Stan: Spatio-temporal adversarial networks for abnormal event detection. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1323–1327. IEEE, 2018. Cited on p. 9.
- [9] Yang Cong, Junsong Yuan, and Ji Liu. Sparse reconstruction cost for abnormal event detection. In *CVPR 2011*, pages 3449–3456. IEEE, 2011. Cited on p. 9.

- [10] VK Gnanavel and A Srinivasan. Abnormal event detection in crowded video scenes. In *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*, pages 441–448. Springer, 2015. Cited on p. 10.
- [11] Cewu Lu, Jianping Shi, Weiming Wang, and Jiaya Jia. Fast abnormal event detection. *International Journal of Computer Vision*, 127(8):993–1011, 2019. Cited on p. 10.
- [12] Tian Wang, Zichen Miao, Yuxin Chen, Yi Zhou, Guangcun Shan, and Hichem Snoussi. Aed-net: An abnormal event detection network. *Engineering*, 5(5):930–939, 2019. Cited on p. 11.
- [13] A Nady, A Atia, and A Abutabl. Real-time abnormal event detection in crowded scenes. *Journal of Theoretical and Applied Information Technology*, 96:6064–6075, 2018. Cited on p. 12.