Contents

Chapter 1

Introduction

Historically, technology has revolutionized police practices. The introduction of the telegraph in the late nineteenth century and the use of two-way radios, motor vehicles and computer-aided dispatching during the twentieth century have brought about dramatic changes in the organizations of police work and, with them, new public expectations of police services.[1] There is, therefore, every reason to expect that the latest round of technological change the information technology revolution will have an equally dramatic impact on policing.[2] Although there is now a growing body of research on technology-based organizational change, the impact of information technology on police practice has not received much research attention.

Observing or analyzing a particular site for safety and business purposes is known as video surveillance. Security and crime control concerns are the motivating factors for the deployment of video surveillance cameras. Video surveillance cameras are used in shopping centres, public places, banking institutions, companies and ATM machines.

Over the last two decades, video surveillance systems have evolved as a result of several changes which have taken place over time:

1. Changing social perceptions and attitudes towards security.
2. Evolving technology platform.
3. Wider range of CCTV users.
4. Wider range of security tasks.

There is no question that video surveillance equipment can be easily and cheaply deployed to monitor practically any environment [3]. Surveillance systems are often ineffective due to insufficient numbers of trained supervisors watching the footage and the natural limits of human attention capabilities [4]. This is understandable, when considering the huge numbers of cameras that require supervision, the monotonic nature of the footage, and the alertness required to pick up on events and provide an immediate response. In fact, even the seemingly simpler task of searching recorded videos, off-line, for events that are known to have happened, requires the aid of Computer Vision systems for video retrieval (e.g., [5]) and summarization[6].

Nowadays, researches experience continuous growth in network surveillance. Therefore, there is a need of a smart surveillance system for intelligent monitoring that captures data in real time, transmits, processes and understands the information related to those monitored. Hence, these systems ensure high level of security at public places which is usually an extremely complex challenge. As video cameras are available at good price in the market, hence video surveillance systems have become more popular. Video surveillance systems have wide range of applications like traffic monitoring [7] and human activity understanding [8].

Benefits of Video Surveillance:

1. Availability- There was a time when the surveillance techniques were utilized only in shopping centres and malls. Now-a-days, you can notice closed-circuit televisions almost at any place you visit, from a small store to homes and holy places. As a result, they guarantee greater public security at a fraction of the cost.

2. Real-time monitoring- Traditionally big organizations have always had the benefits of video surveillance manned by security professionals. In the past times, the events captured on video were used to expose important information and work as proof after the event happened. But, modern technologies let users to check and reply to alarms immediately.

1.1 Motivation

Using a number of video cameras, a large amount of visual data is captured that is to be monitored and screened for intrusion detection. Presently, the surveillance systems used requires constant human vigilance. However, the humans have limited abilities to perform in real-time which reduce the actual usability of such surveillance systems. Also such surveillance systems are not reliable for real time threat detection.

This system can be embedded within a communication system to make a completely autonomous policing system. The law enforcement authority can get alerts of any suspicious activity happening all over the city without any human monitoring each CCTV.

1.2 Objective

The proposed system aims to fulfill the following objectives:

- Overcome limitations of traditional surveillance techniques like human vigilance.
- Increase effectiveness of law enforcement authority by providing comprehensive information about any activity to the nearest police control room, reducing response time as well in the process.

Chapter 2

Related Work

Various techniques[7] have been advocated for detecting and tracking objects in video. Corner and edge features can be clustered together to form objects, and then tracked. Alternatively snake contours can be used to detect an object outline and then track This review is not concerned with biometrics, which involves the identification of individuals using face recognition or some other means. Rather, it focuses on techniques[10] for detecting kinds of objects which are likely to be of interest, such as people, or cars or tanks. Surveillance generally demands that objects are tracked over long periods of time, and in varying conditions. This raises difficulties such as tracking in very different lighting conditions (possibly day and night), across a cluttered and dynamic background, and in the presence of shadows. Because tracking in surveillance video is difficult, on a prior model of the target (such as an articulated human body model, or simple model of a car parameterized on width and height is often used.[11]

An automated surveillance system known as Knight [8] is a fully automated, multiple camera surveillance[9] and monitoring system can detect and classify targets and seamlessly track them across multiple cameras using state-of-the-art computer vision techniques. It also generates a summary in terms of key frames and the textual description of trajectories to a monitoring officer for final analysis and response decision. Current system limitations include the inability to detect camouflaged objects, handling large crowds, and operating in rain and extreme weather conditions.[12]

Among the earlier automated monitoring systems, Pfinder [13] is perhaps the most well known. It tracks the full body of a person in the scene that contains only one unoccluded person in the upright posture. It uses a unimodal background model to locate the moving person. In Rehg et al.,[14] a smart kiosk is proposed that can detect and track moving people in front of a kiosk by using face detection, color, and stereo. Stauffer and Grimson [15] used an adaptive multimodal background subtraction method for object detection that can deal with slow changes in illumination, repeated motion from background clutter, and long-term scene changes. They also proposed detection of unusual activities by statistically learning the common patterns of activities over time. They tracked detected objects using a multiple hypothesis tracker. Ricquebourg and Bouthemy [16] proposed tracking people by exploiting spatiotemporal slices. Their detection scheme involves the combined use of intensity, temporal differences between three successive images, and comparing the current image to a background reference image, which is reconstructed and updated online.

Changes in CCTV technology have led to an increasing number of systems being deployed by public and commercial sector organisations to achieve an increasing number of security tasks. As a result, there is now a wider range of CCTV users, with varying skills and experience. Table 5.2 provides a number of examples in which digital CCTV technology is currently being utilised for different applications by different groups of CCTV users.For e.g.:

1. In the US, web users view live CCTV video via the Internet to monitor the Texas-Mexico border for illegal crossings and alert the authorities (Web Users to Patrol, 2006).

2. In the UK, residents of a housing project view digital CCTV images from their television sets by subscribing to a community safety channel and alert the police by telephone if they witness unusual events or suspicious individuals (Rights Group, 2006).

2.1 Survelliance scenario in India

AI that could thwart illegal activity by identifying criminals before they act is set to be rolled out in India.The aim of the Minority Report-style CCTV surveillance system is to prevent offences such as sexual assault by looking at the body language of people to predict what they are about to do.An Israeli security and AI research company will soon use AI to analyse the terabytes of data streamed from CCTV cameras in public areas in India.The technology will monitor individuals by looking for small twitches that might mean they are about to do something illegal. For example in self-driving taxis the system could detect if someone might be about to assault another person.With crowds, it could also monitor when a situation might be about to turn potentially dangerous.[17]

2.2 Surveillance scenario in China

Facial recognition is the new hot tech topic in China. Banks, airports, hotels and even public toilets are all trying to verify peoples identities by analyzing their faces. But the police and security state have been the most enthusiastic about embracing this new technology.The intent is to connect the security cameras that already scan roads, shopping malls and transport hubs with private cameras on compounds and buildings, and integrate them into one nationwide surveillance and data-sharing platform.[18]

It will use facial recognition and artificial intelligence to analyze and understand the mountain of incoming video evidence; to track suspects, spot suspicious behaviors and even predict crime; to coordinate the work of emergency services; and to monitor the comings and goings of the countrys 1.4 billion people, official documents and security industry reports show.At the back end, these efforts merge with a vast database of information on every citizen, a Police Cloud that aims to scoop up such data as criminal and medical records, travel bookings, online purchase and even social media comments and link it to everyones identity card and face.

With facial recognition we can recognize strangers, analyze their entry and exit

times, see who spends the night here, and how many times. We can identify suspicious people from among the population.

2.3 Surveillance scenario in USA

The various surveillance equipment used in USA are[19]:

Number plate readers

Police cars mounted with automatic number plate readers are thought to be in use in many US cities, gathering data on the location and movements of drivers.

Crime prediction software

Software is being used by police in the US and UK that analyses crime statistics to predict where it will happen next. Microsoft, IBM and Hitachi are among the big players moving into this market. The latest Hitachi "crime visualisation" software - effectively a Domain Awareness Centre on your computer desktop - is being trialled in Washington DC. There is also growing concern about the use of social media analysis software, which monitors hashtags such as BlackLivesMatter and PoliceBrutality to identify "threats to public safety".

Surveillance enabled light bulbs

LED light bulbs marketed as energy-efficient upgrades to existing light bulbs on city streets that can contain tiny cameras and microphones linked to a central monitoring station.

Chapter 3

Proposed Work

In this section, the dependencies from open source image processing libraries, methodology and software constraints are discussed in detail.

3.1 Development Tools Used

OpenCV Open Source Computer Vision (Version 3.3.0)

Open Source Computer Vision, also known as OpenCV, is a real time computer vision library with many image processing functions developed by Intel for the C++ or Java programming platform. This API provides many functions to perform calculation on captured video sources and special image processing on each frame of the video to support this software. We used the python interface of OpenCV.

Flask (Version 0.12.2)

Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. The framework aims to alleviate the overhead associated with common activities performed in Web development . Flask web application code is, in most cases, highly explicit. . It doesn't require a ton of boilerplate (like Django) but has good plugin support. It comes with a basic development server and debugger

Requests (Version 2.18.4)

Requests is a Python HTTP library, released under the Apache2 License. The goal of the project is to make HTTP requests simpler and more human-friendly. It is thread safe and provides proxy support, sessions, authentication and browser style SSL verification. It is most commonly used for web scraping and fetching.

PyQt (Version 5.6.2)

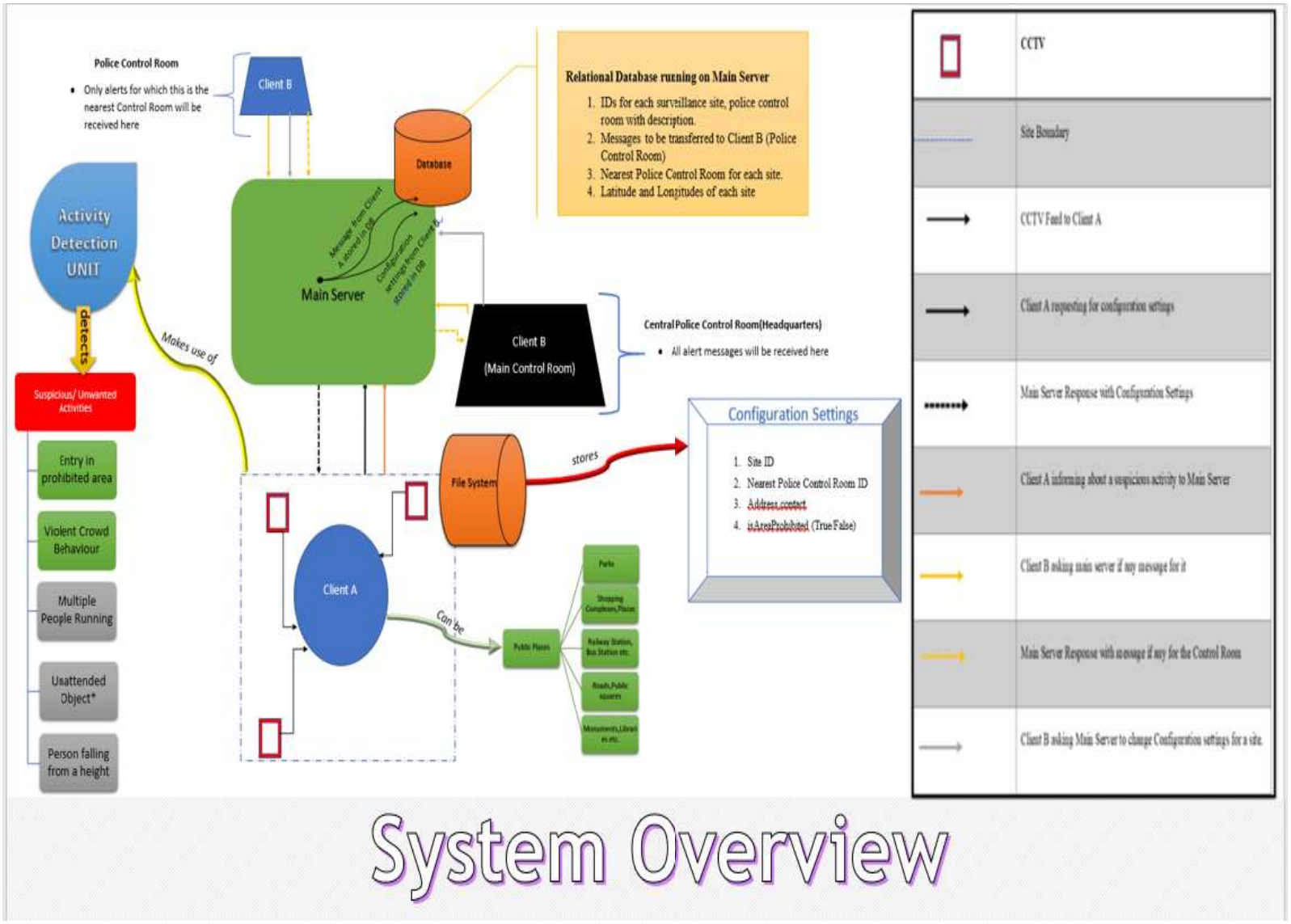
PyQt is a Python binding of the cross-platform GUI toolkit Qt, implemented as a Python plug-in. PyQt5 is the most popular option for creating graphical apps with Python

Google Maps Javascript API

The Google Maps JavaScript API lets you customize maps with your own content and imagery for display on web pages and mobile devices. The Google Maps JavaScript API features four basic map types (roadmap, satellite, hybrid, and terrain) which you can modify using layers and styles, controls and events, and various services and libraries.

3.2 Proposed Methodology

The system would make use of the existing CCTV infrastructure in public places like parks, roads, shopping complexes etc. and detect suspicious activities using sophisticated computer vision techniques. The system would alert the Main Police Control Room (or Headquarter) and the nearest Police Station with relevant information in case any suspicious/unwanted activity is detected.



System Overview

The system consists of two distinct components:

3.2.1 Activity Detection Module

This component detects any suspicious/unwanted activity in real time from the video feed and informs the communication component. The activities the system aims to detect are:

- Pedestrian activity in Prohibited Area
- Violent Crowd Behaviour
- Overcrowding
- Two or more people running (theft)
- Person falling
- Reckless/Drunk/Rash Driving

Currently the system supports the first two activities only.

Pedestrian activity in Prohibited Area

Objective of this part is to detect presence of people (or) a person in a prohibited area and snap a picture of the person entering into the prohibited area using histogram of oriented gradients(HOG).

Histogram of oriented gradients(HOG)

HOG algorithm is used object recognition with very high success rate.We used default Hog descriptor defined in Opencv.
HOGDescriptor(Size win_size=Size(64, 128), Size block_size=Size(16, 16), Size block_stride=Size(8, 8), Size cell_size=Size(8, 8), int nbins=9, double win_sigma

=DEFAULT_WIN_SIGMA,doublethreshold_L2hys=0.2, bool gamma_correction=true, int nlevels=DEFAULT_NLEVELS)
Parameters:

- win_: size Detection window size. Align to block size and block stride.
- block_size : Block size in pixels. Align to cell size. Only (16,16) is supported for now.
- block_stride : Block stride. It must be a multiple of cell size.
- cell:Cell size. Only (8, 8) is supported for now.
- nbins : Number of bins. Only 9 bins per cell are supported for now.
- win_sigma :Gaussian smoothing window parameter.
- threshold_L2hys : L2-Hys normalization method shrinkage.
- gamma_correction : Flag to specify whether the gamma correction preprocessing is required or not.
- nlevels : Maximum number of detection window increases.

cv2.HOGDescriptor() which initializes the Histogram of Oriented Gradients descriptor. Then, we call the setSVMDetector to set the Support Vector Machine to be pre-trained pedestrian detector, loaded via the cv2.HOGDescriptor_getDefaultPeopleDetector() function.
Detecting pedestrians in images is handled by making a call to the **detectMultiScale** method of hog descriptor. The detectMultiScale method constructs an image pyramid with Scale=1.05 and sliding window step size of (4,4) pixels in both the x and y direction, respectively.

HOGDescriptor::detectMultiScale(const GpuMat& img, vector¡Rect¿&, found_locations, double hit_threshold=0, Size win_stride=Size(), Size padding=Size(), double scale0=1.05,

int group_threshold=2)

A larger Scale size will evaluate less layers in the image pyramid which can make the algorithm faster to run. However, having too large of a scale (i.e., less layers in the image pyramid) can lead to pedestrians not being detected. Similarly, having too small of a scale size dramatically increases the number of image pyramid layers that need to be evaluated. Not only can this be computationally wasteful, it can also dramatically increase the number of false-positives detected by the pedestrian detector.

HOG does not work in real time due to its high requirement of processing. So instead of processing every frame, we process those frames in which a motion is detected. For motion detection we use background subtraction algorithm.

BackGround Subtraction algorithm

The simplest method for motion detection is background subtraction. The main advantage of this method is that it is very fast in computation,so suitable for real-time application such as video surveillance.
A motion detection algorithm begins with the segmentation part where foreground or moving objects are segmented from the background. The simplest way to implement this is to take an image as background and take the frames obtained at the time t, denoted by I(t) to compare with the background image denoted by B. Here using simple arithmetic calculations, we can segment out the objects simply by using image subtraction technique of computer vision meaning for each pixels in I(t), take the pixel value denoted by P[I(t)] and subtract it with the corresponding pixels at the same position on the background image denoted as P[B]. In mathematical equation, it is written as:

$$P[F(t)] = P[I(t)] - P[B] \tag{1}$$

The background is assumed to be the frame at time t. This difference image would only show some intensity for the pixel locations which have changed in the two frames. Though we have seemingly removed the background, this approach will only work for cases where all foreground pixels are moving and all background pixels

are static. A threshold "Threshold" is put on this difference image to improve the subtraction (see Image thresholding).

$$|P[F(t)] - P[F(t+1)]| > Threshold \tag{2}$$

This means that the difference image's pixels' intensities are 'thresholded' or filtered on the basis of value of Threshold. The accuracy of this approach is dependent on speed of movement in the scene. Faster movements may require higher thresholds.

Violent Crowd Behaviour

The focus of this part is to monitor crowded events for outbreaks of violence. In order to design a system capable of operating in real time we forgo high-level shape and motion analysis (e.g., [21]) and intensive processing [22], instead following the example of methods for dynamic texture recognition, such as [23], in collecting statistics of densely sampled, low-level features. For the purpose of violence detection in crowded scenes we show that accuracy can be achieved, without compromising processing speed, by considering how flow-vector magnitudes change through time. We collect this information, over short frame sequences, in a representation which we call the VIolent Flows (ViF) de1 scriptor. ViF descriptors are then efficiently labeled as violent or non-violent using a standard linear Support Vector Machine (SVM).

Given a video sequence S of frames (f1,f2,...) We consider two related but different tasks. The first is violence classification: The video S is assumed to be segmented temporally, containing T frames portraying either violent or non-violent crowd behavior. The goal is to classify S accordingly. The second is violence detection: Here, we assume an input stream of frames and the goal is to detect the change from non-violent to violent behavior, with the shortest delay from the time (frame) that the change occurred.

ViF Representation

Given a sequence of frames, S, we produce the VIolence Flows (ViF) descriptor by first estimating the optical flow between pairs of consecutive frames[14]. This provides for each pixel $p_{x,y,t}$ where t is the frame index, a flow vector ($u_{x,y,t}$, $v_{x,y,t}$), matching it to a pixel in the next frame t + 1. Here, we consider only the magnitudes of these vectors:

$$m_{x,y,t} = \sqrt{u_{x,y,t}^2 + v_{x,y,t}^2} \tag{3}$$

For each pixel in each frame we obtain a binary indicator $v_{x,y,t}$, reflecting the significance of the change of magnitude between the frames

$$b_{x,y,t} = \begin{cases} 1 & if\ |m_{x,y,t} - m_{x,y,t-1}| \geq \theta \\ 0 & Otherwise \end{cases} \tag{4}$$

where θ is a threshold adaptively set to the average value of $|m_{x,y,t} - m_{x,y,t-1}|$ Doing so provides us with the binary, magnitude change significant map b_t for each frame f_t. We next compute a mean magnitude change map by simply averaging these binary values, for each pixels, over all frames $f_t \epsilon S$

$$\overline{b_{x,y}} = \frac{1}{T} \sum_t b_{x,y,t} \tag{5}$$

The ViF descriptor is therefore produced by partitioning $\bar{b}$ into M N non-overlapping cells and collecting magnitude change frequencies in each cell separately. The distribution of magnitude changes in each such cell is represented by a fixed-size histogram. These histograms are then concatenated into a single descriptor vector.

Classification with ViF descriptors

For a given sequence S we produce its ViF representation. Each such vector is then classified as representing an either violent or non-violent video. We use simple linear support vector machines (SVM) as the underline classifier. As a consequence, real-time violence detection is achieved by considering short frame sequences, encoding each using its ViF descriptor and then immediately classifying it.

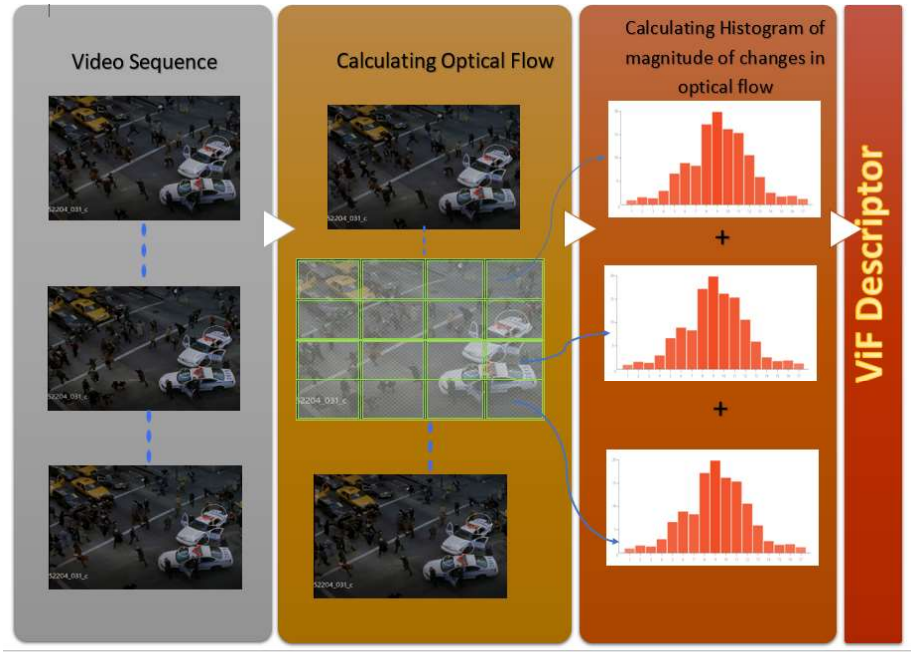


Figure 1: Steps to calculate ViF Descriptor

3.2.2 Communication Component

This component is responsible for transferring of information from the site under surveillance to the pertinent police control room.
The proposed architecture is as follows
There are 3 types of nodes:

Main Server

All communications in the system occur through this server. A simple web server has been utilised for this purpose.A RESTful web service is made to run on the server which responds to the request of all other clients. Currently the built-in server of flask has been utilized.
It also maintains a database which stores:

1. Configuration settings for each surveillance site(Discussed in detail later)
2. IDs for each surveillance site, police control room along with description,address,contact and other relevant info.

3. Messages received from surveillance sites

The particulars of the database schema can be procured from the diagram

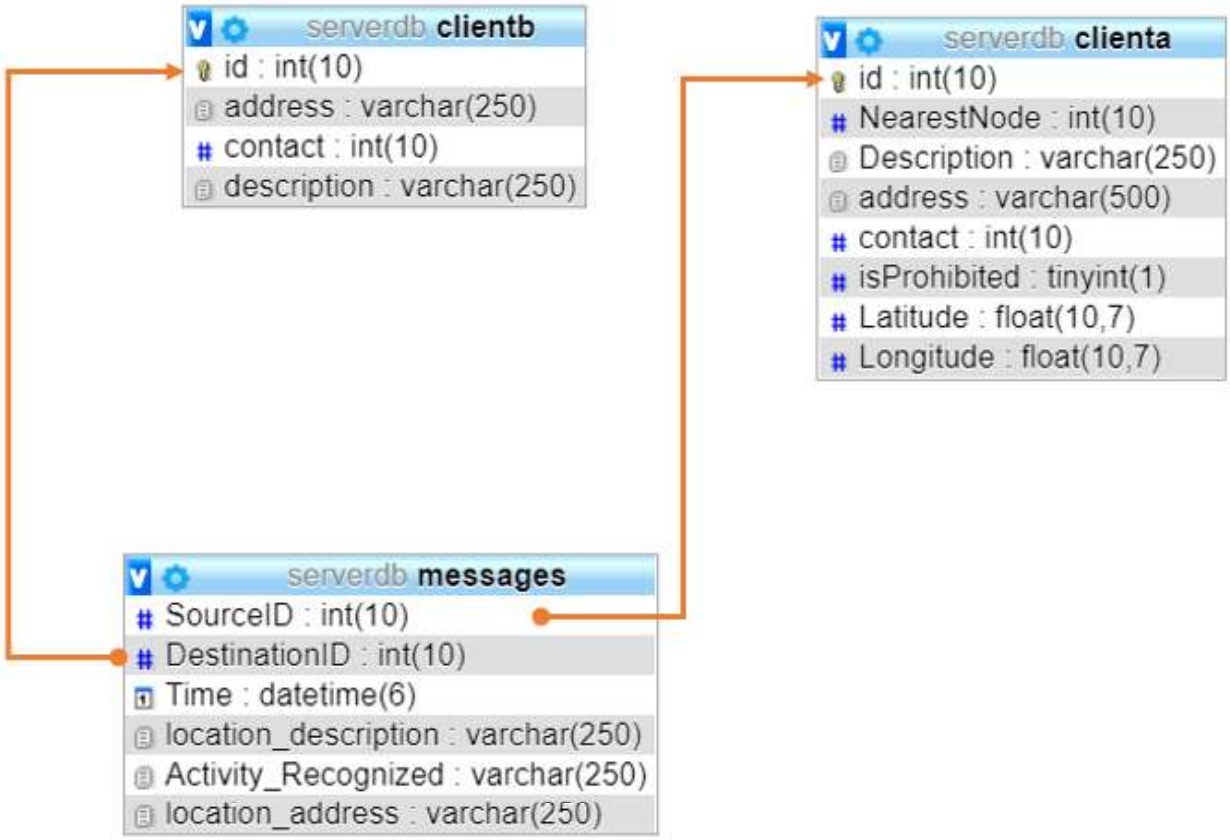


Figure 2: Database Design

Client A

Each site under surveillance will have a node of this type. It will receive video feed from all CCTVs on this site. It will process the feed and send a thorough account of any activity to the main Server.

Each site will have its own configuration settings. It will retrieve these settings periodically from the main server.
The message sent to main server will contain these fields:

1. Site ID
2. CCTV ID
3. Activity Recognized
4. Time Stamp
5. Image Sequence

Configuration settings govern the functioning of this node and also keep details about the node.

These settings will be stored in the local FileSystem itself.

Client B

Each Police Control Room will have a node of this type. It will periodically request the main server (at small interval of 1-2 seconds) to check if it has any new message which is destined for this control Room.
The police headquarter houses a special type of clientB.
It enjoys the following privileges apart from normal nodes:

1. It receives alerts from all surveillance sites while other control rooms receive only if they are the nearest site.
2. It is provided with a GUI which shows which site has reported an alert. The GUI makes it easier and intuitive to get information about any site.

Configuration Settings

These settings enable the server to have fine control over each surveillance site with respect to its functioning and self information.

Configuration Settings include these fields:

1. Site ID
2. Nearest Police Control Room ID
3. Main Police Control Room ID
4. Address,Contact,Description etc.
5. isAreaProhibited (True/False)

Each ClientA node periodically(15 seconds) requests the main server for its configuration settings and works accordingly.
Why is it useful?
The configuration settings empower us to use the same system for all public spaces at all times.This is because for each site we can define its behaviour at all times. For example: There may be some places where the crowd density is usually high during day(metro stations etc.) but there are places where crowd density is low(highways,parks etc.) so we can set the maxCrowdDensity field for each site separately. There are places where multiple people running is not considered a suspicious activity (parks,railway stations etc.) but is suspicious in places like shopping complexes. You might also want to change settings with time of the day, like there should be no human movement inside after a shopping complex or park has been closed or a place has been sealed for investigation.

The typical workflow of the system is as follows:

1. The activity detection unit detects any unwanted activity
2. Client A located at the Site prepares a comprehensive message for the main server and sends it
3. Client A also sends an image of the activity to the server
4. Main server stores the message in its database along with information about its source and destination

5. ClientB periodically request for any messages destined for it and the server obliges.

6. The concerned police control room gets the message and image responds accordingly

3.2.3 Implementation Details

Flask framework was used to create a RESTful web service running on the main server.
All messages are propagated using HTTP POST method. All messages need to be converted into JSON format which is a popular and lightweight syntax for exchanging and storing data.
At client side requests API for python is employed for sending HTTP requests containing data in JSON format and receiving the response from server.

Sending Images

The image is retrieved from the activity detection module and stored locally on clientA in jpg format. The file is given a unique name so that the name can be recreated from message contents.
The name format used is YY_MM_DD_H_M_S_siteID
The image is uploaded to the server as a multipart encoded file.
The server then stores the file locally and keeps track of the images that have been fully uploaded. This is made possible by using consistent filename as described earlier.
When clientB receives a message , it prepares another request to get image from server. It recreates the filename using message content and sends it in request body. The server makes use of Flasks sendfile function to send the image in HTTP response.

Sending Videos

The mechanism for sending videos is pretty similar to sending images. The video codec used is **mpeg** and file format used is **mp4**.The video consists of 15 frames

written at 5 fps.Hence a video of 3 seconds
The media(image and video) evidence allows police personnel to reject any false alarms thereby increasing the effectiveness of the system manifold.

3.2.4 Graphical User Interface(Google Maps)

An intuitive and interactive GUI is developed using Google Maps. It enables the personnel at police headquarters to monitor the complete city. All the surveillance sites are marked on the map. If any suspicious activity is recorded the systems alerts the police personnel and also highlights the location of the activity. The personnel can also seek information of any surveillance site by clicking in the map itself. Google maps JavaScript API is used to embed map in the GUI. Google maps geocoding API is used for conversion between addresses and geolocation. HTML and CSS have been used to customize alert windows in map.QWebView is a WebKit widget from the Pyqt library. It enables us to use JavaScript code from within a python Qt application.
A button is also displayed on the customized info window on Google Maps. The officer can view the video or image as applicable to the activity by clicking on it.

3.2.5 Raspberry Pi

The Raspberry Pi is a credit-card-sized computer that costs between 3000 and 5000 rupees. Its available anywhere in the world, and can function as a proper desktop computer or be used to build smart devices.
Its perfect for projects where you need a computer but dont require much processing power, want to save on space, and keep the costs low.With the small footprint and electricity usage of the Pi, it is a perfect solution for an always on device - meaning you can turn it on, set it up and leave it running somewhere without worrying too much about your electricity bill at the end of the month.
All the above features make raspberry pi ideal for our application. We used Raspberry pi3 model B running on Raspian Stretch for the always-on device needed at surveillance site.

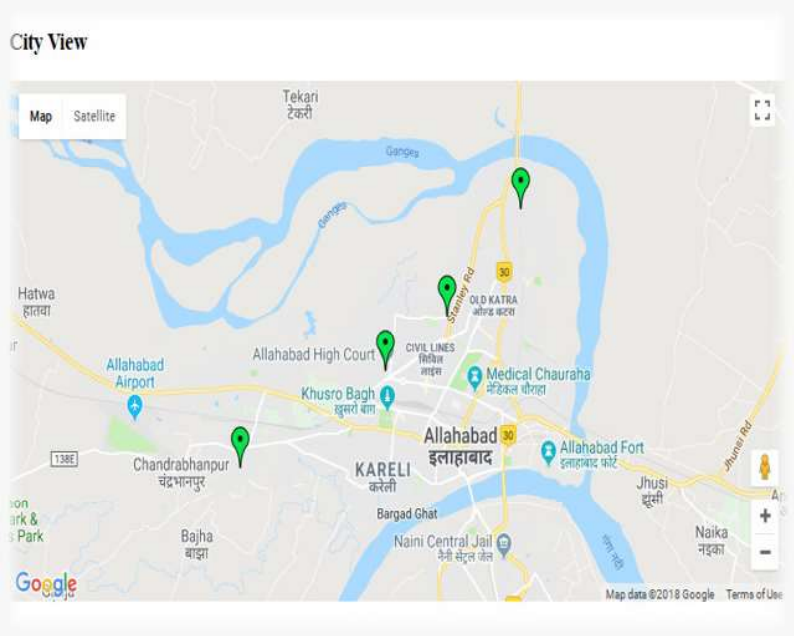


Figure 3: City View.

Figure 4: Suspicious activity detected

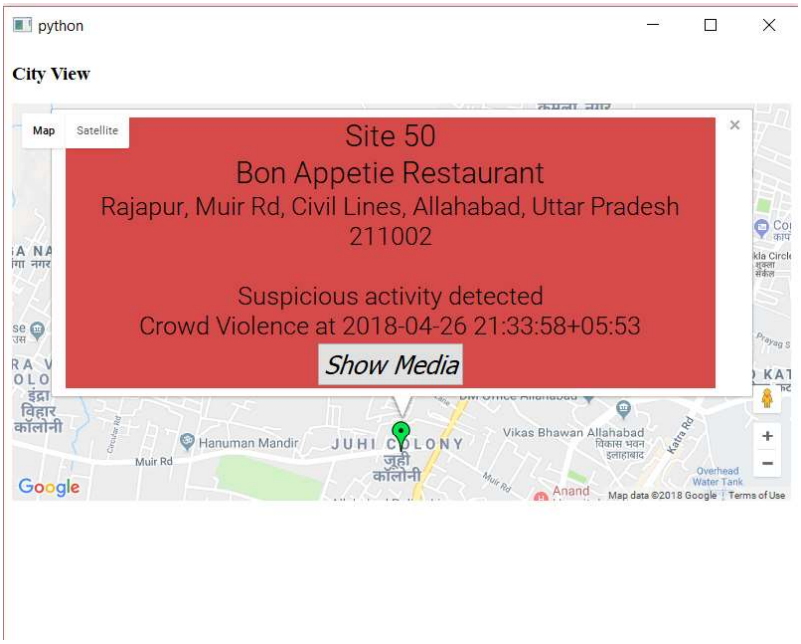


Figure 5: Suspicious site information

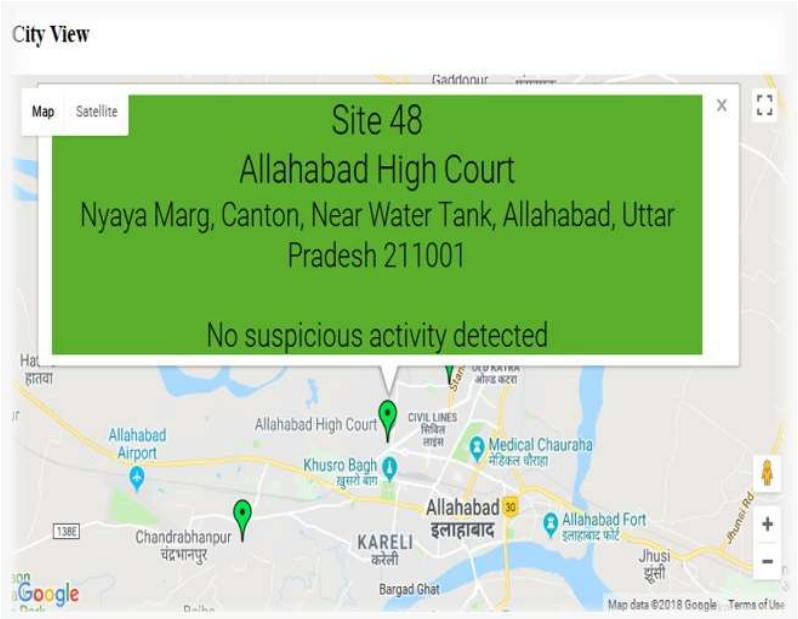


Figure 6: Neutral site information

Chapter 4

Experimental Results and Analysis

The system consists of 3 critical components which need testing and analysis:

1. Activity Detection Module
    - Violent Crowd Behaviour Detection
    - Pedestrian activity in prohibited area
2. Communication Component

We used a desktop PC running on windows 10(i5-4cores CPU 8GB RAM) for Main Server.ClientA and ClientB were implemented on desktop PC running on windows10(i5-2cores 8GB RAM). ClientA was also tested on Raspberry Pi3(1GB RAM) running on Raspian Stretch.

4.1 Results and Analysis

The experimental results and analysis for each of the 3 components is shown in following sections:

4.1.1 Activity Detection Component

Currently the system supports detection of two activities. Each activity was tested under different benchmarks as applicable.

- *Human detection in prohibited area*

Human detection in prohibited area is done with HOG descriptor algorithm and with some optimizations.

S.No	WinStride(x,y)	Processing time(per frame)
1	4,4	0.47sec
2	8,8	0.10sec
3	16,16	0.071sec

S.No	ScaleFactor	Processing Time(per frame)
1	1.01	0.50sec
2	1.06	0.10sec
3	1.3	0.03sec
4	1.5	0.029sec

Figure 7: Results of parameter tuning in HOG descriptor

Thresholding Area parameter used in background subtraction also effects in processing time. If thresholding area is less false positive increases thereby increasing processing time and vice verse.

Figure 8: Detection Results

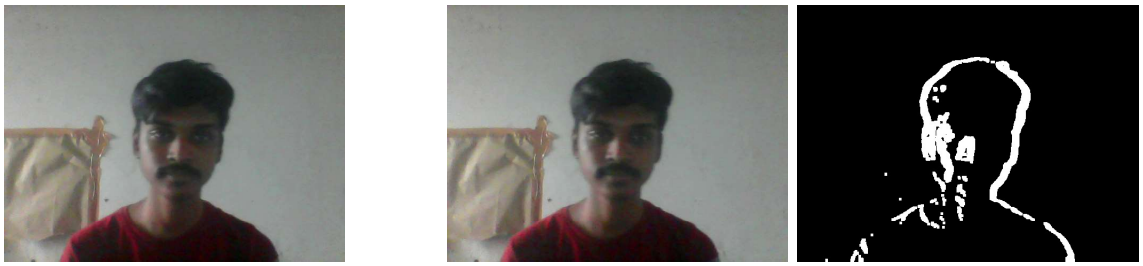


Figure 9: Motion detection using Background subtraction

Inference: There is trade off between accuracy and speed if we want to make HOG to run in real time. The accuracy with small values of Winstride(x,y) and ScaleFactor is high with a downside of speed. With smaller values of Winstride(x,y) and ScaleFactor it has higher processing time. So, the optimal values of Winstride(x,y)-(8,8) and ScaleFactor=1.06 are taken. The images in which Pedestrians are far off from view are not detected

Violent Crowd Behaviour Detection

The algorithm was implemented in python.We made use of optical flow code available from [24] and linear SVM from scikit.
We used a grid size of M*N=4*4.To achieve real time performance we processed 1 in every 3 frames for accurate temporal detection. Each sequence of 15 frames(after sampling)is classified as violent or non-violent using SVM.
Violent-Flows database is an evaluation benchmark for crowd violence detection.There are 246 videos in this database (123 violence and 123 non-violence). All the videos are downloaded from the web with average 3.60 seconds and are under uncontrolled, in-the-wild conditions.Five-fold cross validation is a common validation manner which is also adopted here in experiments. The 246 videos are almost equally distributed in 5 parts, each part containing equal number of violent and non-violent videos randomly selected.The first four sets are used for training and fifth is used for testing purposes. The training set contains 46 videos of which 23 depict violence and 23 depict non-violence.The results can be visualized from the confusion matrix in Figure 11.

Figure 10: Violent Activity Detection Results

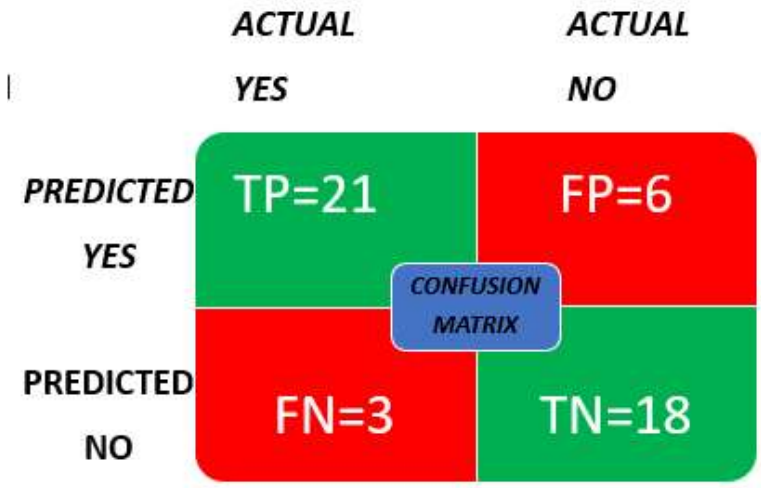


Figure 11: Confusion Matrix for Crowd Violence detection model

TP:True Positive
FP:False Positive
FN:False Negative
FP:False Positive

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 81.25\% \tag{6}$$

$$Sensitivity = \frac{TP}{TP + FN} = 87.5\% \tag{7}$$

Inference: The ViF descriptor designed for crowd violence detection proposed in [15] has been used for classification. The accuracy achieved is good considering the complex nature of crowd movements. Though an algorithm with higher accuracy would be needed for the purpose of our system. It can be inferred that the ViF descriptor is unable to capture some important information. Another important parameter is sensitivity in case of our system. A high sensitivity means a very low probability of any violent activity going unnoticed. Hence sensitivity is most important for our system. We have laid more stress on increasing sensitivity as false positives can easily be ruled out using media evidence.

4.1.2 Communication Component

3 clientA systems and 3 clientB systems were made to run simultaneously with input from locally stored videos. A private network was needed for communication. The system was tested using both mobile hotspot and college network. No faults/data loss in Communication system were observed during testing. There were no errors or atypical behaviour detected in the communication component. There was a maximum lag of 3 seconds between detection of activity at surveillance site and message alert at control room.

Inference:

Due to robust and uncomplicated software design we were able to trace down and weed out the bugs in the communication system. All the message formats used had minimal size and hence put less load on the network. The media from a CCTV camera were of low size, hence did not take much time to be transferred over the network.

4.2 Performance

4.2.1 Activity Detection Component

We assume an input stream of frames and the goal is to detect the change from normal to suspicious behavior, with the shortest delay from the time (frame) that the change occurred. We consider a delay of 6-7 seconds acceptable for a detection system

- *Pedestrian activity detection in prohibited area*

HOG takes a lot of computation for human detection. It takes 0.1 seconds to process each frame. So, instead of processing each frame we process only the frame in which motion is detected. Now a lot of processing can be skipped and thus making it feasible to run in real time.

- *Violent activity detection*

The performance results are presented in figure 11. We process selective short frame sequences separately, classifying each one as either violent or nonviolent; a detection is reported once a violent sub-sequence of frames is thus encountered. Acceptable results were obtained when we used PC for processing, though the performance on raspberry pi were below par. Extraction of feature vector(ViF descriptor) for a sequence of 15 frames took on an average 5 seconds to be processed. Hence appropriate sampling had to be done without compromising on delay in alerts. Experiments showed that not more than 15 continuous frames are required for classification.

	Raspberry Pi	PC
Pedestrian Detection	4.1	2.6
Violent Crowd Detection	5.1	4.3

Figure 12: Average delay between time of occurrence of activity and raising of alert at clientB(in seconds)

4.2.2 Communication Component

Even with 6 clients running simultaneously there was no observant lag in transferring of information from one node to other. There was an average delay of 3 seconds between detection of activity and message reaching the control room.

4.3 Objectives Achieved

The following objectives have been achieved so far:

1. The complete architecture for communication has been designed and implemented. The communication component is working smoothly with low latency.
2. Image and Video of the activity was also obtained at Police Control Room from surveillance site.
3. Pedestrian detection module has been integrated with the communication component and is also giving acceptable results.
4. Violent Crowd detection module has been integrated with the communication component .

4.4 Deployment

The system will make use of existing CCTV infrastructure. For development purposes we have used Flasks inbuilt server. Though it performs good enough in development stages but is not suitable for production as it does not scale and provides minimal security features. It is highly recommended to use a real web server from reliable hosting service providers like AWS Elastic Beanstalk,Microsoft Azure. Proper authentication measures would also be need to set up when the system goes online. Two feasible approaches have been chalked out for deployment at surveillance site.The choice of approach depends on type of surveillance needed on site, number of CCTVs and availability of resources.

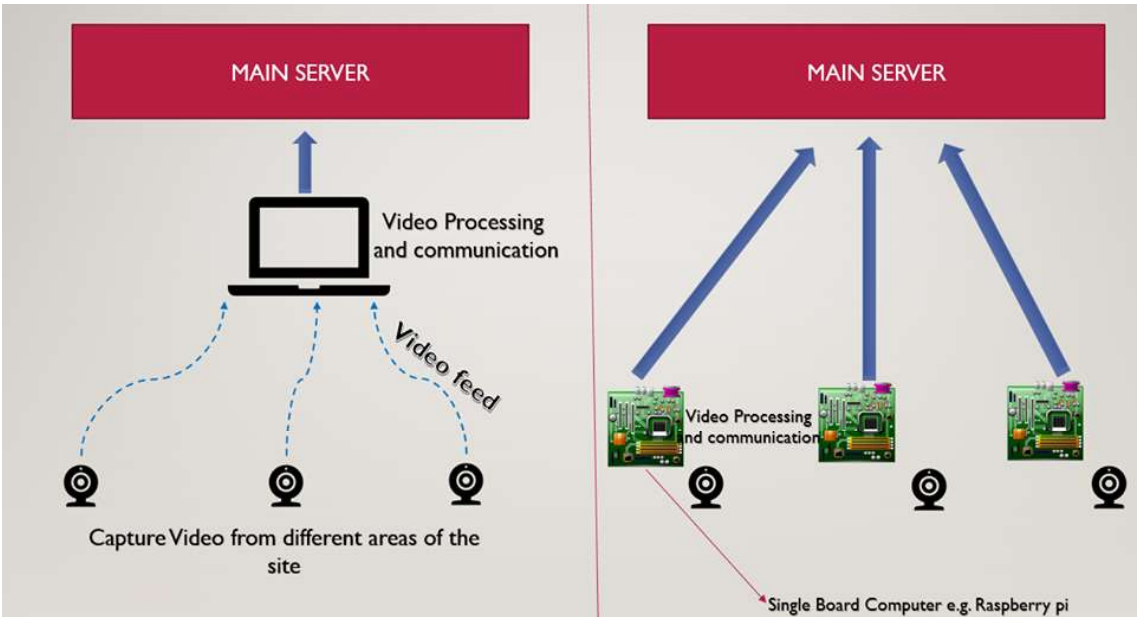


Figure 13: Two feasible approaches for deployment at surveillance site

Chapter 5

Conclusion and Future Work

Despite recent progress in computer vision and other areas, there are still major technical challenges to be overcome before the dream of reliable automated surveillance is realized. These technical challenge include considerations such as the physical placement of cameras, the network bandwidth required to support them, installation cost, privacy concerns and robustness to unfavorable weather and lighting conditions. However progress is being made ever more rapidly, and the demand for automated surveillance continue to increase in areas ranging from crime prevention, public safety and home security to industrial quality control and military intelligence gathering.

The system proposed is performing well and giving acceptable results. The architecture is designed in such a way that it can be easily scaled and extended for additional features. The communication component is based on REST architecture. The system provides close control over the behaviour of surveillance sites remotely through configuration settings. Pedestrian Movement Detection has been achieved with help of HOG and background subtraction has been additionally added to provide near Real-Time performance. Violent Crowd behaviour Detection needs to be improved further to be used real-life scenarios.The results from this algorithm were below acceptable levels.

5.1 Difficulties and Problems Faced

1. Transferring of images was a critical task for the system. The transferring of image is remarkably different from transferring messages. The speed of transfer of messages is naturally much faster than transferring of image file. Hence it was difficult to synchronize.

2. Achieving real time performance is crucial need for the system.It was a huge challenge to perform the analysis of video in real time.

3. Embedding a city-wide map using javascript in a python desktop application posed numerous roadblocks.

5.2 Future Work

The system offers a host of bright prospects for extending the work to make it even more effective:

1. Detecting other unwanted activities like abnormal crowd density, multiple people running (theft etc.), person falling etc. from CCTV footage in real time.

2. GUI dashboard for controlling settings for each site and CCTV.It can be made available to the police headquarters.

3. Sending video sequence of activity to police headquarter

4. The headquarters should be able to request video clip of any particular time from any site.

5. If a police station is unable to receive alerts, send the alert as a text message to the phone no.

6. Create a network of police control rooms, so that if the nearest control room is not functioning the message should be passed to next nearest node.

   This would replace existing infrastructure and make it easy to deploy the proposed system.

References

[1]Abt Associates 2000, Police Department Information Systems Technology Enhancement Project (ISTEP), Department of Justice, Office of Community Oriented PolicingServices, Washington, DC.

[2]Bingham, M. 1996, Report on the Review of the Queensland Police Service, Brisbane

[3]T. Hassner, Y. Itcher, and O. Kliper-Gross, Violent Flows: Real-Time Detection of Violent Crowd Behavior, 3rd IEEE International Workshop on Socially Intelligent Surveillance and Monitoring (SISM) at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Rhode Island, June 2012 .

[4]H. Keval. Effective, design, configuration, and use of digital CCTV. PhD thesis, University College London, 2009.

[5]N. Petrovic, N. Jojic, and T. Huang. Adaptive video fast forward. Multimedia Tools and Applications, 26(3):327344, 2005.

[6]Y. Pritch, S. Ratovitch, A. Hendel, and S. Peleg. Clustered synopsis of surveillance video. In Advanced Video and Signal Based Surveillance, pages 195200, 2009.

[7] M. Shah, O. Javed, K.Shafique,Automated Visual Surveillance in Realistic Scenarios, IEEE MultiMedia Vol. 14, Issue: 1, 2007

[8]O. Javed, K. Shafique, and M. Shah, A Hierarchical Approach to Robust Background Subtraction Using Color and Gradient Information, Proc. IEEE Workshop on Motion and Video Computing, IEEE CS Press, 2002, pp. 22-27.

[9]O. Javed "Tracking across Multiple Cameras with Disjoint Views" ¡em¿Proc. 9th IEEE Int'l Conf. Computer Vision¡/em¿ pp. 343-357 2003.

[10]R. Collins, A. Lipton, T. Kanade, "Introduction to the Special Section on

Video Surveillance", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 745, 2000.

[11]Manohar Karki, Saikat Basu, Robert DiBiano, Supratik Mukhopadhyay, Jerry Weltman, Malcolm Stagg, "A symbolic framework for recognizing activities in full motion surveillance videos", Computational Intelligence (SSCI) 2016 IEEE Symposium Series on, pp. 1-7, 2016.

[12]N. Dalal, B. Triggs, Histograms of oriented gradients for human detection Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE

[13]C. Wren et al., Pfinder, Real-Time Tracking of the Human Body, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 780-785.

[14]J. Rehg, M. Loughlin, and K. Waters, Vision for a Smart Kiosk, Computer Vision and Pattern Recognition, IEEE Press, 1997, pp. 690-696

[15]C. Stauffer and W.E.L. Grimson, Learning Patterns of Activity Using Real-Time Tracking, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 8, 2000, pp. 747-757.

[16]Y. Ricquebourg and P. Bouthemy, Real-Time Tracking of Moving Persons by Exploiting Spatiotemporal Image Slices, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 8, 2000, pp. 797-808.

[17] http://www.dailymail.co.uk/sciencetech/article-5603367/AI-studies-CCTV-predict-crime-happens-rolled-India.html

[18] https://www.washingtonpost.com/news/world/wp/2018/01/07/feature/in-china-facial-recognition-is-sharp-end-of-a-drive-for-total-surveillance/?noredirect=onutm$_t erm$ = .13c86680a705

[19] http://www.bbc.com/news/magazine-37411250

[21]Abdelkader, Mohamed F., et al. "Silhouette-based gesture and action recognition via modeling trajectories on Riemannian shape manifolds." Computer Vision and Image Understanding 115.3 (2011): 439-455.

[22]O. Kliper-Gross, T. Hassner, and L. Wolf. One shot similarity metric learning for action recognition. Similarity-Based Pattern Recognition, pages 3145, 2011.

[23]V. Kellokumpu, G. Zhao, and M. Pietikainen. Human activity recognition using a dynamic texture based method. In BMVC, pages 110, 2008.

[24]C. Liu. Beyond Pixels: Exploring New Representations and Applications for Motion Analysis. PhD thesis, Massachusetts Institute of Technology, May 2009.