ANJUMAN - I - ISLAM'S KALSEKAR
TECHNICAL CAMPUS, NEW PANVEL
ENGINEERING & TECHNOLOGY
ARCHITECTURE
PHARMACY
AKTC
NAVI MUMBAI - INDIA

A PROJECT REPORT

ON

"VIOLENCE DETECTION SYSTEM"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

Sayed Najneen Fatma Mustaq Ali Parveen Fatima	**15CO07**
Chougle Zaid Noornabi Zerin	**15CO15**
Kothari Hozefa Abbas Jumana	**15CO24**

UNDER THE GUIDANCE OF
PROF. Mubashir Khan

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 & 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206 **2018-2019**

AFFILIATED TO

UNIVERSITY OF MUMBAI

A PROJECT II REPORT
ON

"VIOLENCE DETECTION SYSTEM"

Submitted to
UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

Sayed Najneen Fatma Mustaq Ali Parveen Fatima	**15CO07**
Chougle Zaid Noornabi Zerin	**15CO15**
Kothari Hozefa Abbas Jumana	**15CO24**

UNDER THE GUIDANCE OF
PROF. Mubashir Khan

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 & 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2018-2019
AFFILIATED TO

UNIVERSITY OF MUMBAI

Anjuman-i-Islam's Kalsekar Technical Campus

Department of Computer Engineering
SCHOOL OF ENGINEERING TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

CERTIFICATE

This is certify that the project entitled

"Violence Detection System "

submitted by

Sayed Najneen Fatma Mustaq Ali Parveen Fatima	**15CO07**
Chougle Zaid Noornabi Zerin	**15CO15**
Kothari Hozefa Abbas Jumana	**15CO24**

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engineering) at ***Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai*** under the University of MUMBAI. This work is done during year 2018-2019, under our guidance.

Date: / /

(Prof. Mubashir Khan)
Project guide

(Prof. Kalpana Maher Bhodke)
Project Coordinator

(Prof. Tabrez Khan)
HOD, Computer Department

DR. ABDUL RAZAK HONNUTAGI
Director

External Examiner

Acknowledgements

We would like to take the opportunity to express my sincere thanks to our guide **Mubashir Khan**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout our project research work. Without his kind guidance & support this was not possible.

We are grateful to him for his timely feedback which helped us track and schedule the process effectively. His time, ideas and encouragement that he gave is help us to complete our project efficiently.

We would like to express deepest appreciation towards **DR. ABDUL RAZAK HONNUTAGI**, Director, AIKTC, Navi Mumbai, **Prof. Tabrez Khan**, Head of Department of Computer Engineering and **Prof. Kalpana Maher Bhodke**, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped us directly or indirectly during this course of work.

Sayed Najneen Fatma Mustaq Ali Parveen Fatima

Chougle Zaid Noornabi Zerin

Kothari Hozefa Abbas Jumana

Project I Approval for Bachelor of Engineering

This project entitled ***Ÿiolence Detection System¨*** by ***Sayed Najneen Fatma Mustaq Ali Parveen Fatima, Chougle Zaid Noornabi Zerin and Kothari Hozefa Abbas Jumana*** is approved for the degree of ***Bachelor of Engineering in Department of Computer Engineering.***

Examiners

1. .....
2. .....

Supervisors

1. .....
2. .....

Chairman

.....

Declaration

We declare that this written submission represents my ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Sayed Najneen Fatma Mustaq Ali Parveen Fatima
15CO07

Chougle Zaid Noornabi Zerin
15CO15

Kothari Hozefa Abbas Jumana
15CO24

ABSTRACT

As action recognition problem is becoming a hot topic within computer vision, the detection of fights or in general aggressive behavior has been comparatively less studied. Such capability may be extremely useful in some video surveillance scenarios like in prisons, psychiatric centers or even embedded in camera phones

Our System will be capable enough to detect aggressive behavior in a fight and various objects used in it. It will be using various computer vision technique and deep neural network to detect and recognize objects and actions in the violent outbreak. Our System is mainly dependent on how the model is trained and how good the quality of CCTV camera is, as we will be doing the processing on CCTV feed

Keywords: Computer-vision, neural network, violent outbreak, action recognition, object detection, Keras, Numpy, Tensorflow, surveillance video, HMM(Hidden Markov Model)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Our project promises to replace the ongoing manual work of violence detection with an automated approach using deep learning and computer-vision techniques. It will detect the violence from the incoming CCTV feed and will provide us with the output whether it is violent or not

1.1 Purpose

As today surveillance system is managed manually with a compulsory assistance of human interaction but is has been witnessed that a person gets bored or tend to loose interest after performing a specific task for a long period of time and constantly gazing at screen for whole day, 7-days a week can be quite boring. The job of surveillance is quite important when it comes to situations of violent outbreak so keeping this in mind and considering its importance we thought of automating the surveillance system with minimum human interaction and maximum positive output.

1.2 Project Scope

Our system is named as Violence Detection System, the name itself gives a basic gist of what it will be doing. It will detect object and recognize actions but only those which are violent, fascinating isn't it. It will be loaded with computer vision techniques to detect objects like guns, knives etc and neural networks to recognize violent action actions. Basic idea behind using neural networks is that it provides maximum accuracy among-st other machine learning techniques

1.3 Project Goals and Objectives

1.3.1 Goals

- Successfully classify violent and non-violent behavior
- Notify the responsible authority about violent outbreak

1.3.2 Objectives

- To identify the violent behaviour
- To store detailed information of the violent outbreak to ease up the investigation.

1.4 Organization of Report

Chapter **Introduction** shows how this idea popped up and motivation we got to develop this project. We checked if there any system exist for this problem. We found paper based and computer based system. We studied their advantages, disadvantages and got to know how we can build solution to overcome those disadvantages.

Chapter **Literature Survey** includes summary, advantages, disadvantages and ways we can improve those disadvantages of reference paper we studied. Review of literature helps to understand need of project, how project can improve situations and it helps developers to understand what exactly need to develop. Literature review helps clients to know in what areas project can be used.

Project Planning and SRS chapter is given so that other developers or clients can know what technologies, tools, software and hardware is used. On what hardware or platform developed project can be deployed. The market potential of project, its estimated development cost, expected profit can be known from this chapter.

System design chapter is provided with six diagrams to understand modules, users and architecture of project. Use case diagram is given to understand functionality of a system with users and usecases. To visualize database ER diagram is shown. Class diagram is provided to understand structure of project and to understand how data is passing through modules Data Flow Diagram(DFD) is given. To show relation between different modules Component Diagram is shown.

Chapter **Implementation** describes each and every module of project in details. Also to understand interaction logic between object in system sequence diagram

is shown. Activity diagram shows control flow from one activity to another. Flow chart for every module is given that shows overall structure of the process or system, traces the flow of information and work through it, and highlights key processing and decision points.

Chapter **System Testing and Screenshots of project** discusses Test cases used for testing the system, to check validation. The results occurred are given in this chapter. The analysis done after development is described here.

Last chapter **Conclusion and Future Scope** describes how we can make project scope more broad. What are the limitations of system and conclusion.

Chapter 2

Literature Survey

2.1 Violence Detection in Video based on Deep Learning

2.1.1 Summary:

- Each video is framed as RGB images
- After extracting frames from the videos CNN technique is used to detect actions from the frames.
- If actions are detected then those actions are classified using deepCNN and Temporal Segment Networks[1].

2.1.2 Advantages:

- Successfully detect the videos as violent and non violent.
- Improved the accuracy by 97.06%.

2.1.3 Disadvantages:

- Training data is augmented
- missed alarm

2.1.4 How to overcome in our system:

- Train the data in less amount of time
- Alarm will send as soon as the violent video is identified

2.2 Video based Abnormal Human Behavior Recognition

2.2.1 Summary:

- This paper has broadly described about the various algorithm that can be used for various human behavior recognition using surveillance video camera feed.
- It also states about the human behavior in less and more crowded place and how to classify the violent and non-violent video by depicting the data extracted representing in models, there exists wide discussion about HMM(Hidden Markov Model) to generate models for the same.

2.2.2 Advantages:

- It successfully classifies the videos.

2.2.3 Disadvantages:

- A different state space must be manually defined for each different scenario.
- False Alarm was generated.

2.2.4 How to overcome in our system:

- No false Alarm.

2.3 Violent Flows: Real-Time Detection of Violent Crowd Behavior

2.3.1 Summary:

- Firstly it divides the incoming video into sequence of frames(snippets) Then those snippets are used for detecting violent outbreak in crowded scenes
- It has introduced a new technique named Vif and compares it with SVM and other classifiers .

2.3.2 Advantages:

- Vif technique is used for classification of violent behavior which is better than various existing classification .
- Sequences of videos are classified within half a second to be precise 0.28s.

2.3.3 Disadvantages:

- Uses a five cross-validation test

2.3.4 How to overcome in our system:

- Validation will be done by the classifier.

2.4 Technical Review

Our application is fabricated with following technologies: Web technologies, Python, neural networks, tensorflow, opencv . All the listed technologies are described further

2.4.1 Web technologies

Web Technologies is a combination of many languages like HTML, CSS, Js, SQL, PHP, Bootstrap etc.

Reason to use web technologies

- Web Technologies are programming languages such as HTML and CSS, which are well known among IT professionals.
- They run on the device's own web browser through a simple URL.
- It run on any operating system.
- Dynamic and Interactive Web pages.
- Responsive Websites.

2.4.2 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Reason to use python

- Simple Syntax.
- Easily Understandable.
- Extensive Support Libraries.
- Integration Feature.

2.4.3 Neural Networks

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.

Reason to use Neural networks

- It have the ability to learn and model non-linear and complex relationships.
- After learning from the initial inputs and their relationships, it can infer unseen relationships on unseen data as well, thus making the model generalize and predict on unseen data.
- It has the ability to learn hidden relationships in the data without imposing any fixed relationships in the data.

2.4.4 Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks

Reason to use Tensorflow

- The TensorFlow library can work for multifarious perceptual and language understanding tasks and to conduct complicated research on Machine Learning and Deep Neural Networks.
- It performs numerical computations through data flow graphs.
- TensorFlow is highly parallel and designed to use various backends software .
- Scalability.

2.4.5 OpenCV

OpenCV is a library of programming functions mainly aimed at real-time computer vision.OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

Reason to use opencv

- It is use for image processing purpose.
- Ease of use.
- Portability.
- Speed.

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Zaid	UI Design, Php
2	Najneen	Python, UI Design
3	Hozefa	Python, Database

3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Zaid	Co-Team Leader	UI Design
2	Najneen	Co-Team Leader	Algorithm Design
3	Hozefa	Co-Team Leader	Algorithm Design

3.3 Assumptions and Constraints

3.3.1 Assumptions:

Training plays an important role when it comes to deploying a deep learning model. For a fact deep learning model requires a large amount of dataset to get user satisfied accuracy

3.3.2 Constraints:

The video which will be feed-ed should be clear, so that violence can be detected accurately .

3.4 Project Management Approach

We have used Spiral methodology for the development of this project. Spiral Model is a combination of a waterfall model and iterative model. Each phase in spiral model begins with a design goal and ends with the client reviewing the progress. The services provided provided by spiral model of software development satisfies to the dynamic change in requirement of the system. This is the main reason why we chose spiral model for software development as it grants various services like determine objectives, Identify risks, development and test, plan the next iteration and these services can be changed according to the changing environment.

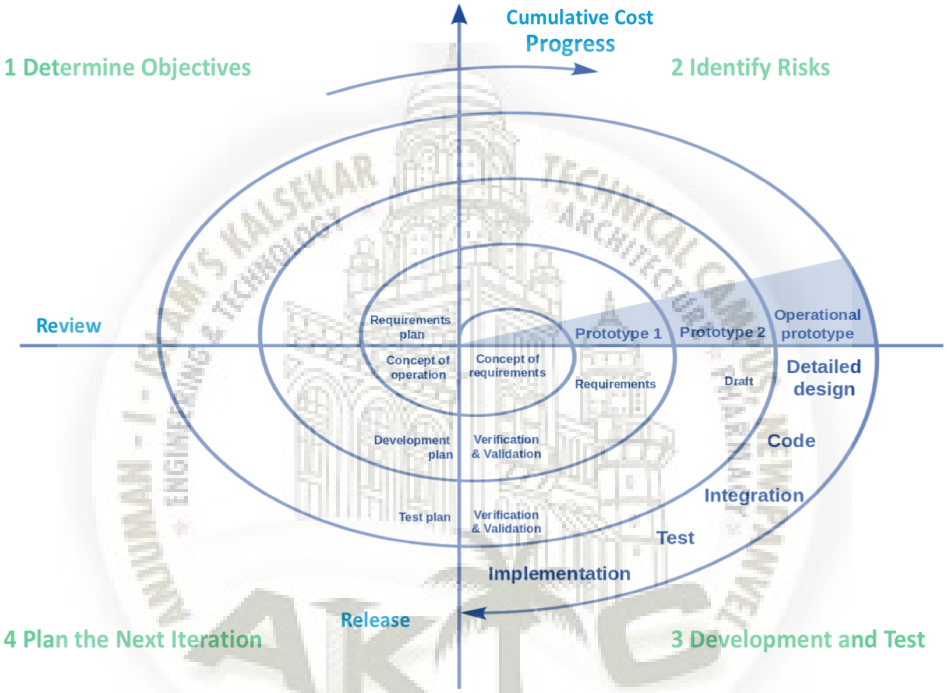


Figure 3.1: Spiral Model

3.5 Ground Rules for the Project

We Consider the following ground rules:

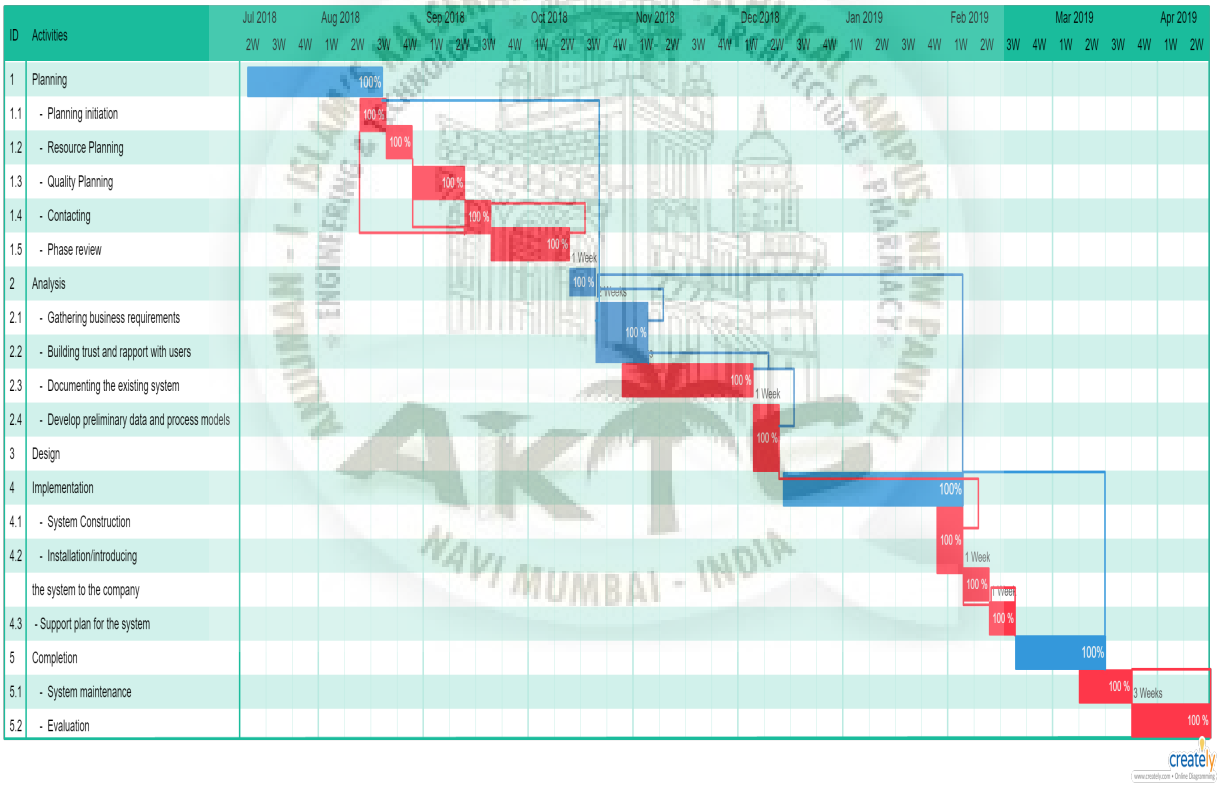
1. Project should also be build from users prospective.
2. We will keep positive attitude towards Project and team members and everyone will respect each other.
3. Everyone will take initiative by sharing ideas telling improvements in each other.
4. We will be honest and take our responsibility , we will try our best to complete our project before deadline .

5. If any member got stuck at something he or she should ask for help to one another.

3.6 Project Budget

- Web technologies : Open Source
- Python : Open Source
- OpenCV : Open Source
- Tensorflow : Open Source

3.7 Project Timeline



Chapter 4

Software Requirements Specification

4.1 Overall Description

4.1.1 Product Perspective

As today surveillance system is managed manually with a compulsory assistance of human interaction so our system will replace the ongoing manual work into a automated one. This system is one of a kind as no one has ever tried before to deploy a system of this magnitude. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.

4.1.2 Product Features

The system will have the capability of classifying the video into the class of violent or not. The system will also detect a probable object that can be used during violent outbreak. Our System is one of a kind. The admin of our system will be overlooking the detected violent data to avoid any miscalculation or false alarm.

4.1.3 User Classes and Characteristics

As our product comes under digital surveillance based system no ordinary user can access the resources without authentic credentials. Only users with high level access and admin privileges can access this system

4.1.4 Operating Environment

The product is purely based on python and its object detection and action recognition libraries, so to avoid any errors the system in which our product would be deployed must attain all the dependencies. Product will be using web technologies to display the necessary output.

4.1.5 Design and Implementation Constraints

As we are trying to depict how a human brain would react after seeing some violent actions in a product. Our product is based on video processing and to do video processing it requires a GPU which comes at a great cost, without that the training part becomes a bit difficult. Secondly being a web application internet facility must be 24x7

4.2 System Features

1. Classification of violent and non violent behavior
2. Object Detection
3. Alert of violent outbreak through system

4.2.1 Classification of violent and non violent behavior

Description and Priority

The primary and most important task of the product is to classify whether the data given is tagged as violent or not

- Stimulus:Needy do login and select the video to detect violent behavior
- Response: Shows whether it is violent or not

4.2.2 Object Detection

Description and Priority

Objects that are used/carried during violent outbreak are also being detected

4.2.3 Alert of violent outbreak through system

Description and Priority

If any violent scenes are captured/detected/recognized a system alert is generated to notify the user that violence is underway

4.3 External Interface Requirements

4.3.1 User Interfaces

- All users should register first to get all services.
- All the data asked in forms should be accurate to get best results.
- To fill form properly hints and tooltips are added to every input field.

4.3.2 Hardware Interfaces

- PC with 8 GB RAM
- Mininum of 4 GB of Nvidia GeForce GPU
- 1280 x 800 minimum screen resolution.
- 2.3 GHz Fast processor.

4.3.3 Software Interfaces

- Web technologies
- Internet Connection.
- Python

4.3.4 Communications Interfaces

- As data flowing is mainly dependent on the user log-in and in our system only high level users are authorized to access the system to maintain the security of the system
- The detection and recognition of the data provided starts after a validated and authorized user log-in
- Classification is done on the basis of the action and object detection and recognition
- After successfully detection and recognizing violent outbreak a alert is generated by the system

4.4 Nonfunctional Requirements

4.4.1 Performance Requirements

The performance of our system is based on how accurate the prediction is done using the data provided. More the data more will be the accuracy

4.4.2 Safety Requirements

Password verification is provided while registering onto the system. The password entered can be helpful in logging in . Hence only genuine people can get access to system.

4.4.3 Security Requirements

Fake users cannot use our system because before entering into system user need to enter authorized credentials as it comes under security domain

Chapter 5

System Design

5.1 System Requirements Definition

To classify the feed into violent and non-violent class we went through many technologies like neural networks and so on. We also studied how human action can be captured and recognized using different computation tools

5.1.1 Functional requirements

- The detection and recognition of the data provided starts after a validated and authorized user log-in
- Classification is done on the basis of the action and object detection and recognition
- After successfully detection and recognizing violent outbreak a alert is generated by the system

Use-case Diagram

.jpg

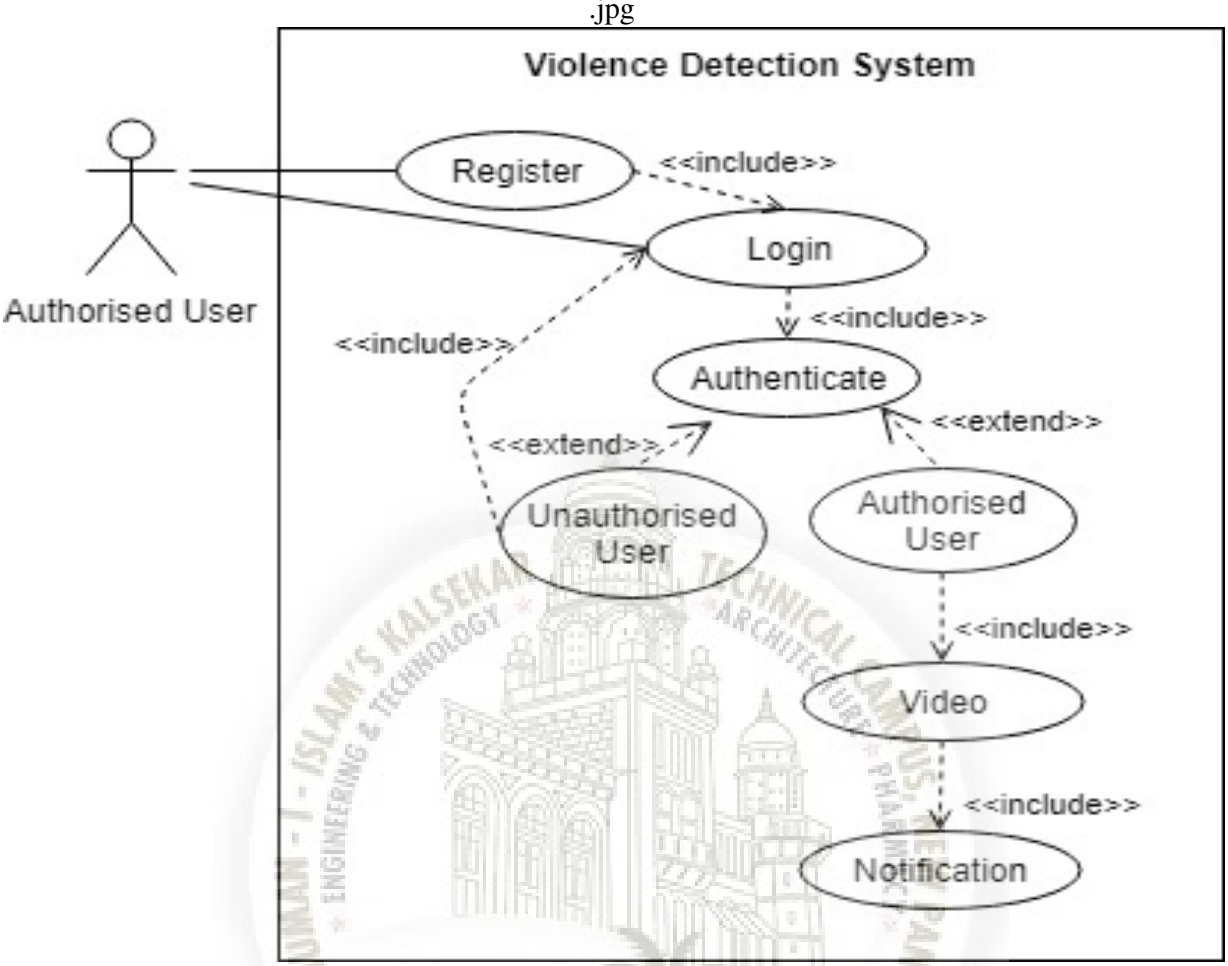


Figure 5.1: Use-Case

Use case diagram is given to understand functionality of a system with users and usecases .

- Firstly , User will Register or log in .
- After that, user will be authenticate ,for this we will use SHA-256 Algorithm.
- As the user is authorized he/she will be able to see the classification of the data which is going on and a notification will be triggered if any violent behaviour is detected.
- If the user is not authenticated then he/she have to log in again.

5.2 System Architecture Design

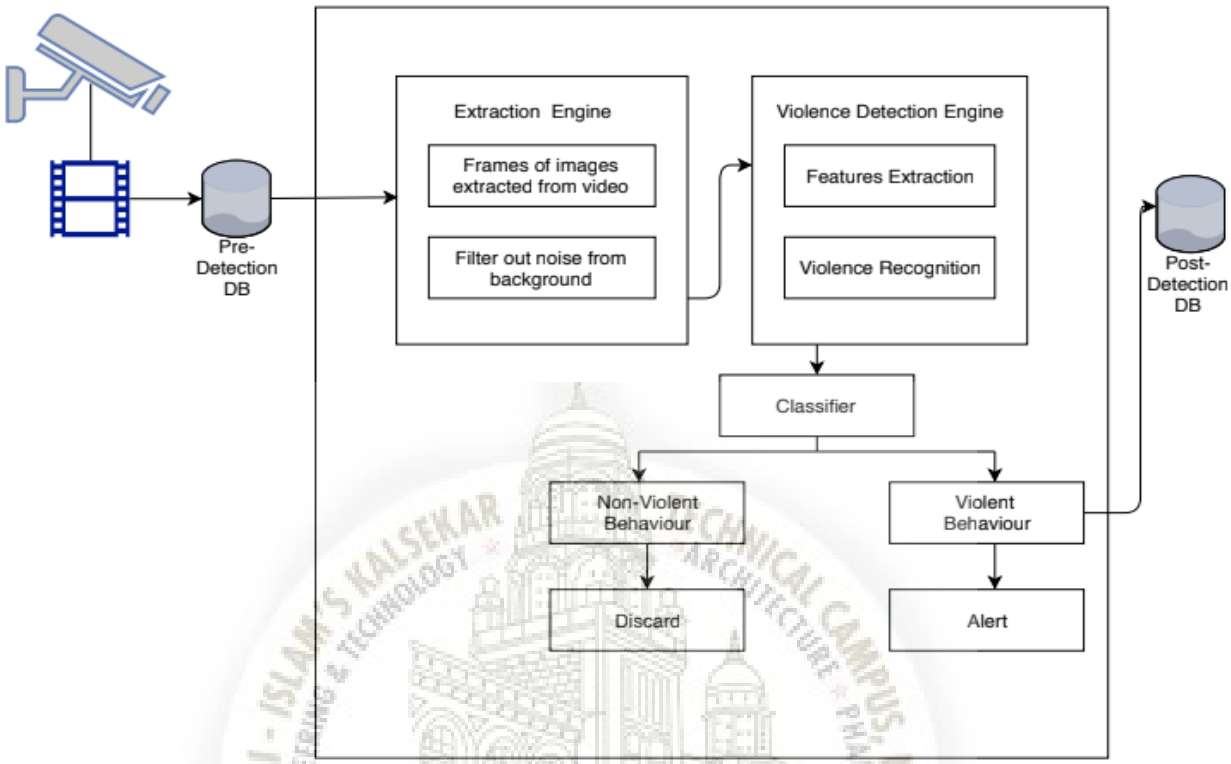


Figure 5.2: System Architecture

- Video file will be taken as an input which will be processed. While processing it will go to Extraction Engine module. Here , it will divide the videos into frames and then denoising of that frames will be done.
- After denoising it will move towards Violence Detection Engine. Here, it will extract the features from frames then according to that violence behaviour will be detected.
- If violent behaviour is detected then a system alert will be generated.

5.3 Sub-system Development

Mainly there are two modules in our system viz., Extraction Engine and Violence-detction engine

5.3.1 Extraction Engine

- First to convert the video into frames of images we will be using OpenCV which is python library.
- Later on Denoising of the frames is done using OpenCV library.
- OpenCV(Open Source Computer Vision Library) is designed for computational efficiency and with a strong focus on real-time applications.
- The library can take advantage of multi-core processing.

Modular Diagram of Extraction Engine

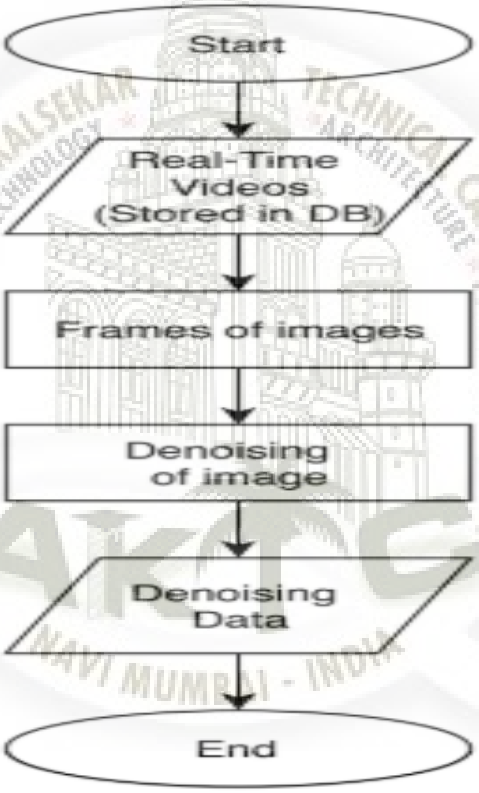


Figure 5.3: Extraction Engine

5.3.2 Violence Detection Engine

- After the denoising of frames is done features are extracted.
- After that, classification of the frames is done using tensorflow
- Tensorflow is a library in python used for action recognition and object detection
- If violant frames are recognized a system alert is popped and the user is notified about the violent outbreak

Modular Diagram of Violence Detection Engine

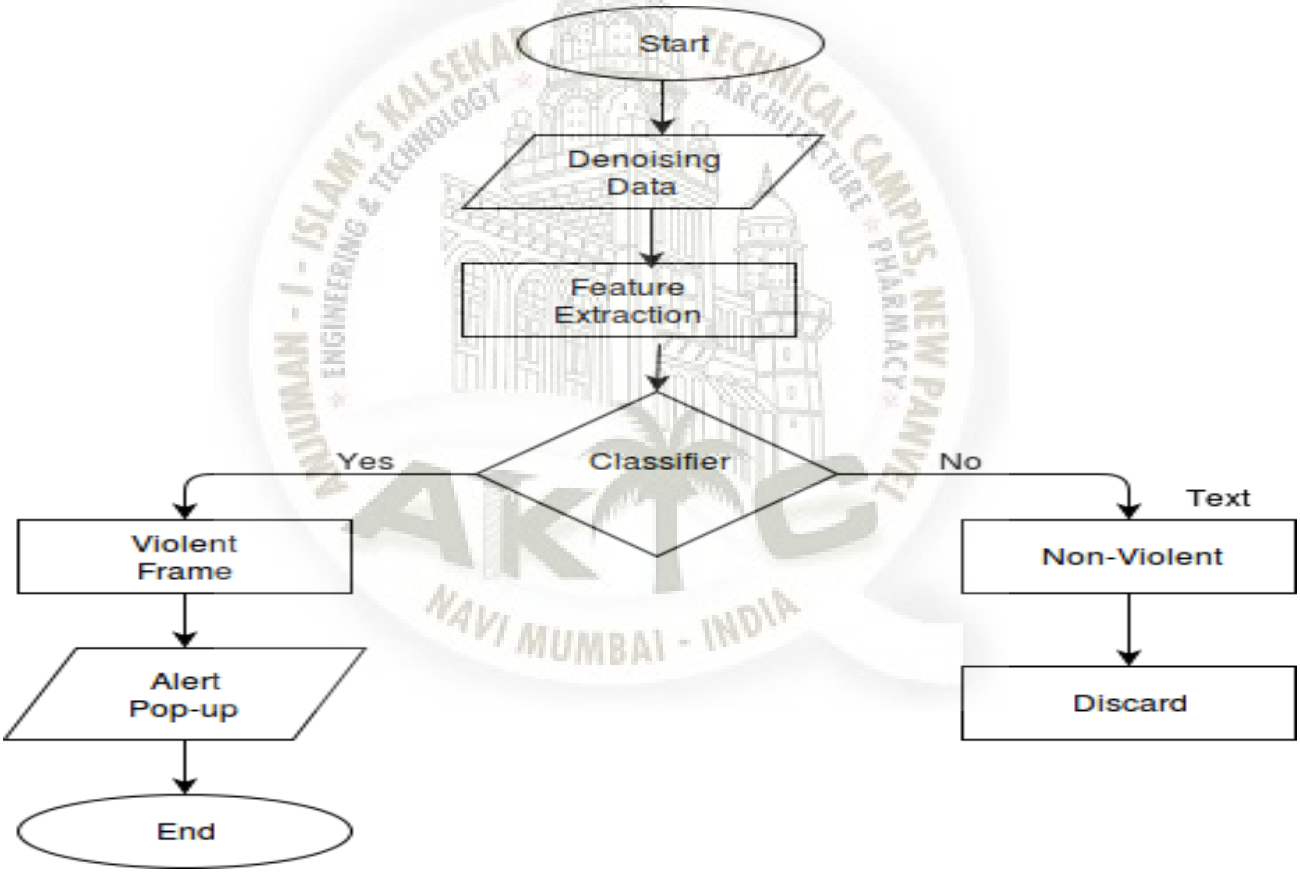


Figure 5.4: Violence Detection Engine

5.4 Systems Integration

In order to achieve goal of system the developed modules need to be get integrated with one another.The Extraction Engine module should work in reconnaissance with the Violence detection module and after this grouping alert should be generated

5.4.1 Class Diagram

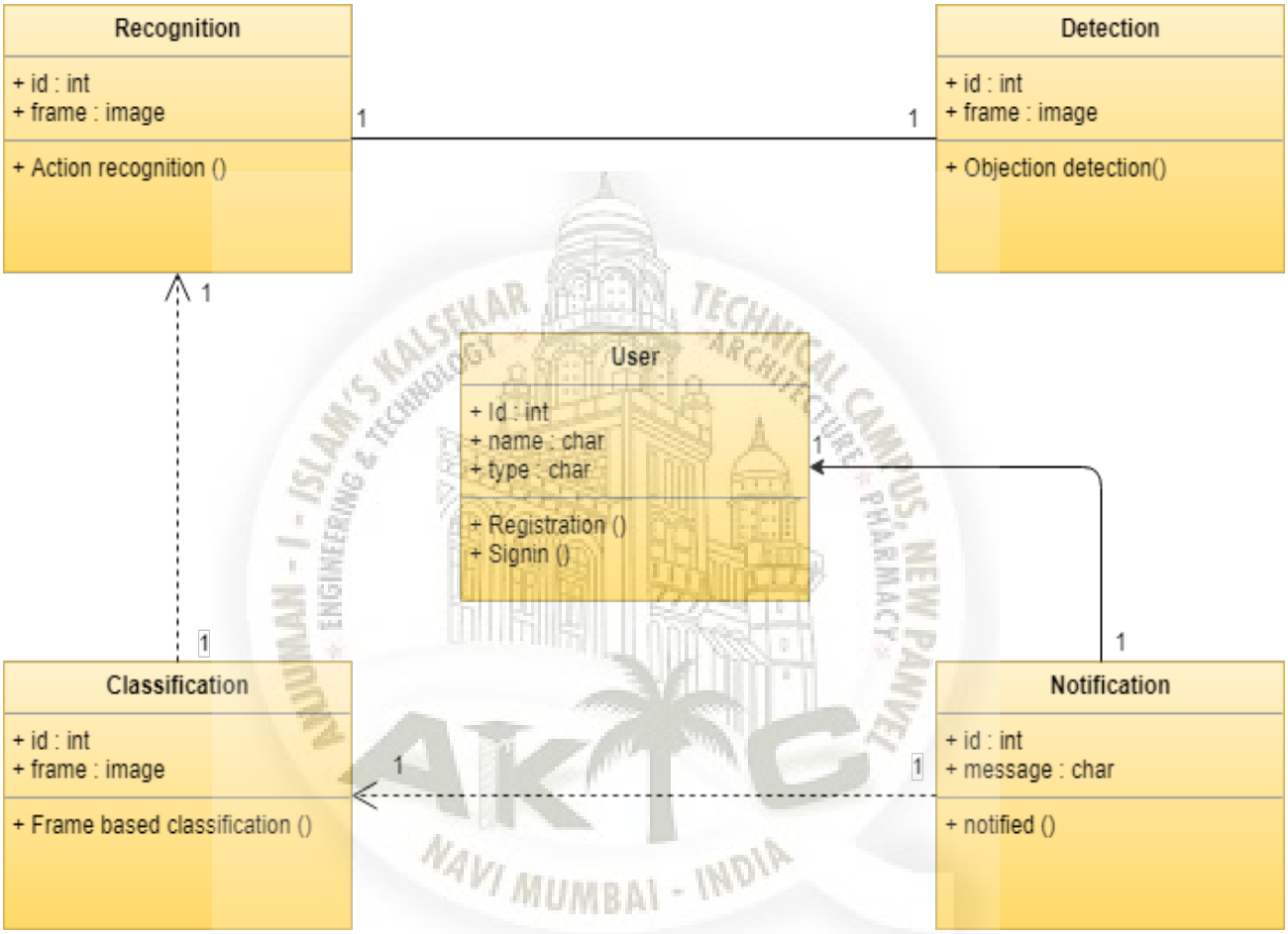


Figure 5.5: Class Diagram

Class diagram gives attributes, operations of module. This class diagram is an illustration of the relationships and source code dependencies among classes in our system. In our system there are this classes:Recognition, Detection, User, Classification, Notification.Recognition and detection classes are dependent on each other and classification class is used for detection and recognition

5.4.2 Data Flow Diagram

Data flow diagram explains how data is transferred through system. Data from which module flowing where can be recognized by this diagram. Data flow diagram helps to identify inputs, outputs for modules.

- DFD Level 0 : It contains total no of 3 process in our DFD level 0 diagram. It has VDS, Data server and Admin .

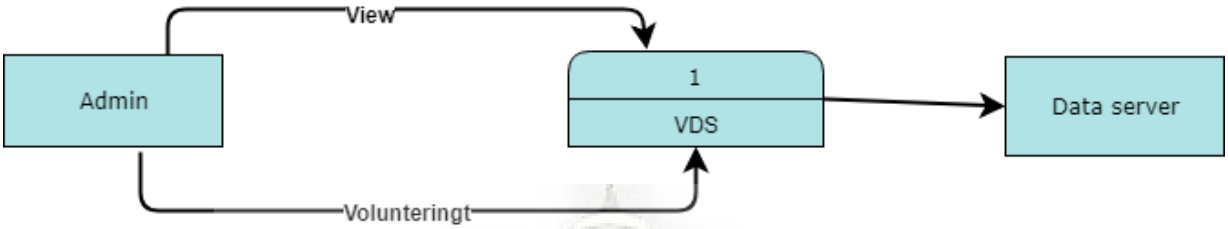


Figure 5.6: DFD level 0

- DFD level 1 : In this level the flow of data from various entities of the system. Data is authenticated and extraction and detection is done which is visible to both Admin and normal user

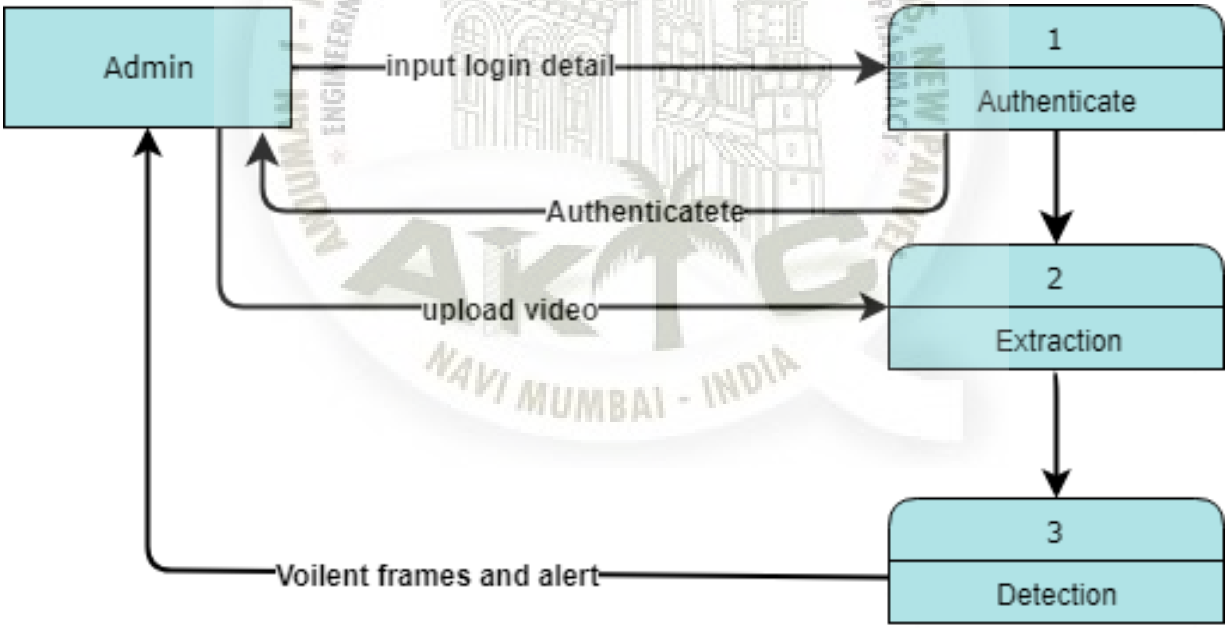


Figure 5.7: DFD level 1

5.4.3 Sequence Diagram

Sequence diagrams are a kind of interaction diagram, because they describe how and in what order a group of objects works together.

User Sequence Diagram

In this sequence diagram is requesting the data to perform detection and the result is being notified to the user

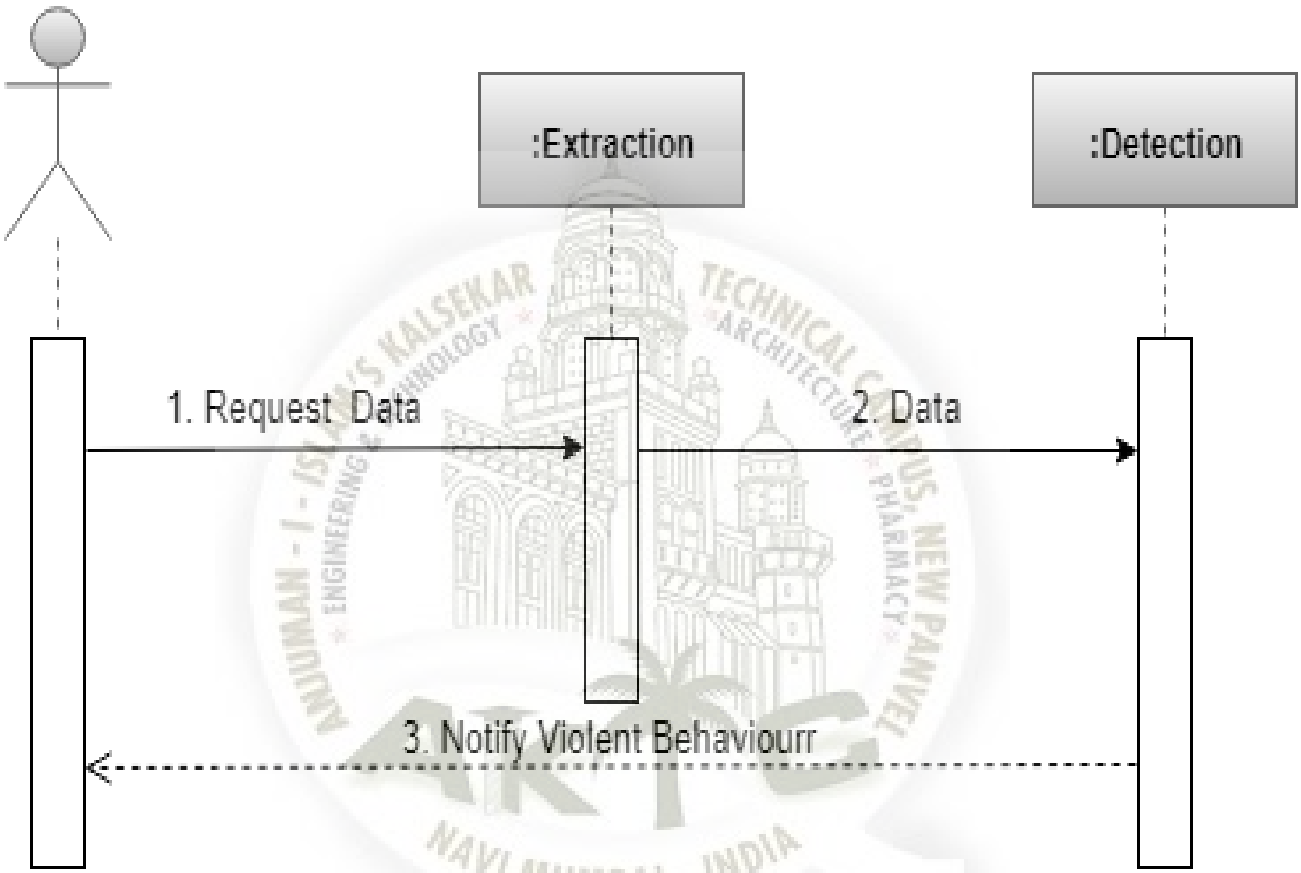


Figure 5.8: User Sequence Diagram

Admin Sequence Diagram

This sequence diagram depicts the role of admin in the system. He/she overlooks the classification in case the classification is done wrong admin notifies the user

Sequeence Diagram.png

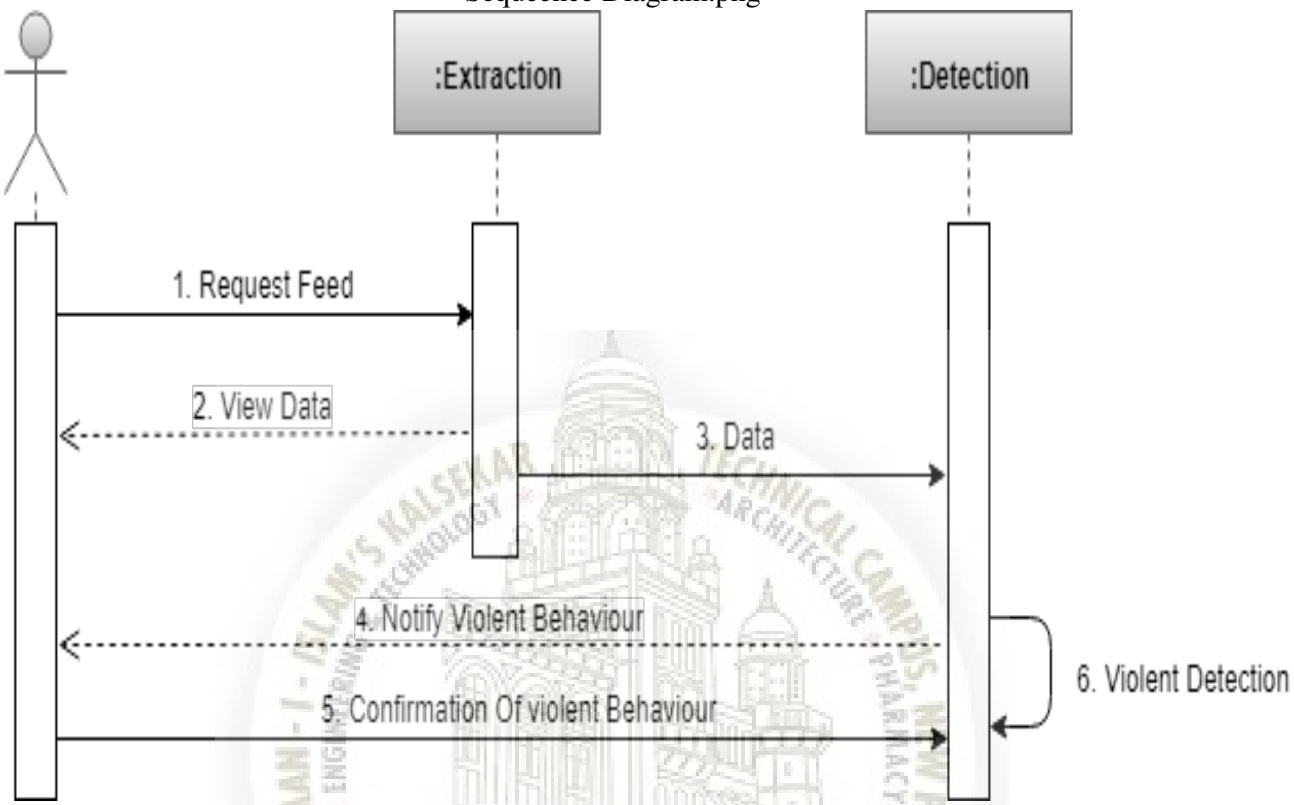


Figure 5.9: Admin Sequence Diagram

5.4.4 Component Diagram

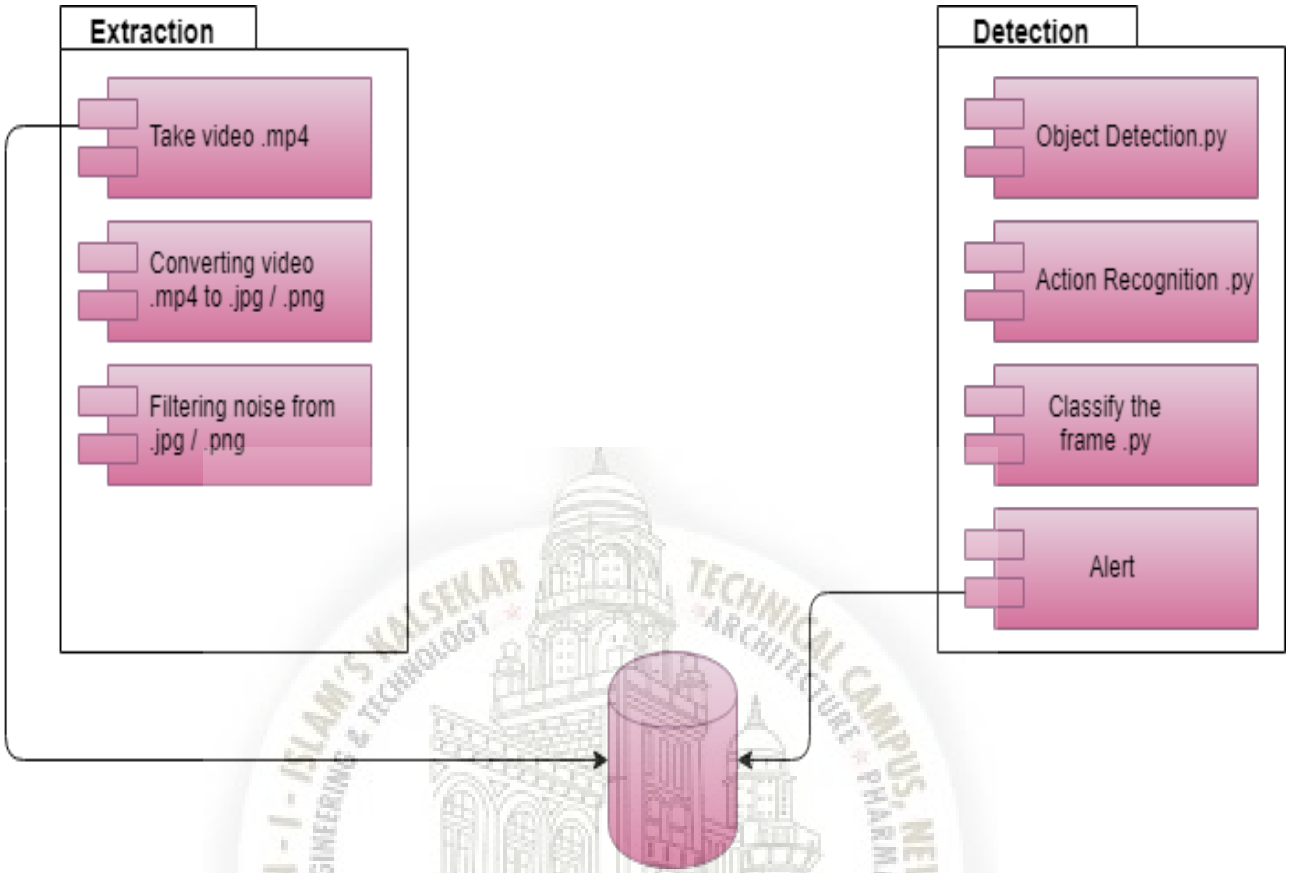


Figure 5.10: Component Diagram

Component diagram shows how all component of system is connected. Component is nothing but subsystem.In our system registration component is necessary for all user. All the details of users are stored in database. Data uploaded in Upload module is stored in Uploads database. Extraction and Detection are the two wheels in the system, in the absence of either of the module the system would fail

5.4.5 Deployment Diagram

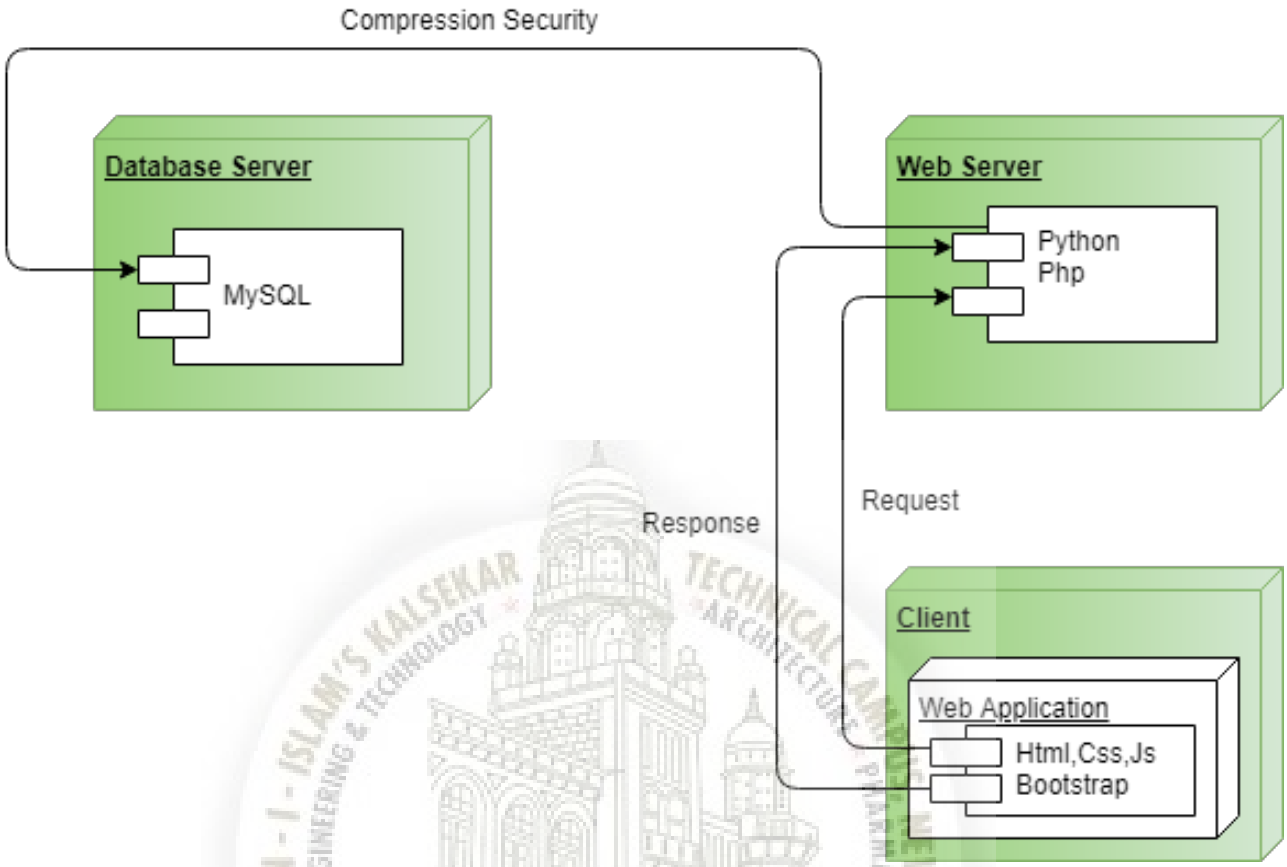


Figure 5.11: Deployment Diagram

Using deployment diagram you can understand how the system will be physically deployed on the hardware.A deployment diagram is a diagram that shows the configuration of run time process. We are using web technologies to deploy our system, as our system consists of python programming web services seems to be the best fit

Chapter 6

Implementation

6.1 Log-In

In order to check whether a particular feed is violent or not the user must first login into the system

Figure 6.1: Log-In Page

```
<?php?>
<link rel="stylesheet" href="loginstyle.css" type="text/css">

<form action="index.php" method="POST">
  <div class="login">
    <div class="login-screen">
      <div class="app-title">
        <h1>V.D.S.</h1>
       </div>

       <div class="login-form">
         <div class="control-group">
         <input type="text" class="login-field" value="" placeholder="username"
             name="username">
         <label class="login-field-icon fui-user" for="login-name"></label>
         </div>

         <div class="control-group">
         <input type="password" class="login-field" value="" placeholder="
             password" name="password">
         <label class="login-field-icon fui-lock" for="login-pass"></label>
         </div>

         </div>
                 <input type="submit" href="index.php" value="Log in" class="btn
                     btn-primary btn-large btn-block" >
           <br>
       </div>
    </div>
  </div>

</form>
```

6.2 Extraction and Detection Engine

Here we are taking an input of video file and we are extracting frames and sending it to the next module

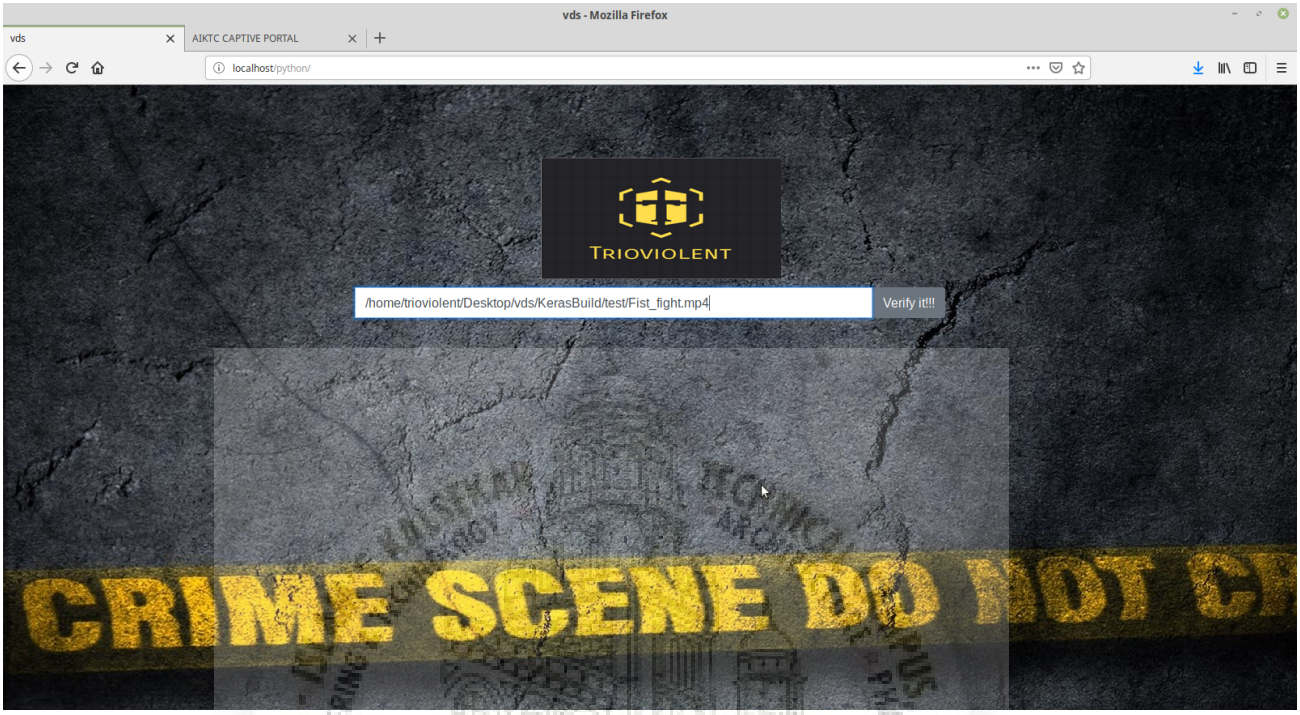


Figure 6.2: Fist Extraction

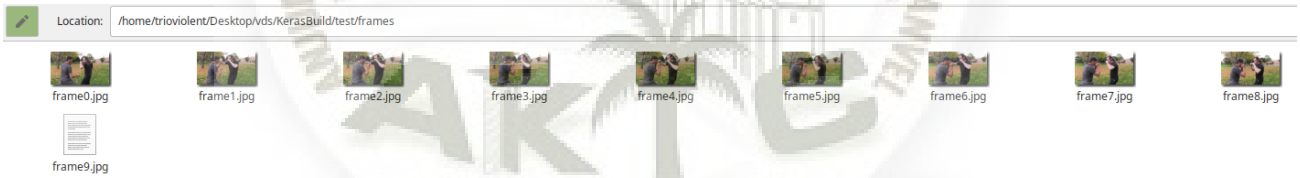


Figure 6.3: Fist Extraction Frames

```
1
2  import cv2
3  import tensorflow as tf
4  import sys
5  import os
6  import csv
7  import json
8
9  vidcap = cv2.VideoCapture('test/Fist_fight.mp4')
10 success,image = vidcap.read()
11 counts = 0
12
13 while success:
14   vidcap.set(cv2.CAP_PROP_POS_MSEC,(counts*1000))
15   success,image = vidcap.read()
16   cv2.imwrite("test/frames/frame%d.jpg" % counts, image)
17   counts += 1
18
19 # Loads label file, strips off carriage return
20 label_lines = [line.rstrip() for line
21                    in tf.gfile.GFile("/home/trioviolent/Desktop/vds/KerasBuild/
                       tf_files/retrained_labels.txt")]
22
23 res = { 'guns' : [] , 'fist fight' : [] , 'knife' : [] }
24
25 for i in range(0,counts-1) :
26   # change this as you see fit
27   image_path = "test/frames/frame"+str(i)+".jpg"
28   # Read in the image_data
29   image_data = tf.gfile.FastGFile(image_path, 'rb').read()
30
31   # Unpersists graph from file
32   with tf.gfile.FastGFile("/home/trioviolent/Desktop/vds/KerasBuild/tf_files/
       retrained_graph.pb", 'rb') as f:
33       graph_def = tf.GraphDef()
34       graph_def.ParseFromString(f.read())
35       _ = tf.import_graph_def(graph_def, name='')
36
37   with tf.Session() as sess:
38       # Feed the image_data as input to the graph and get first prediction
39       softmax_tensor = sess.graph.get_tensor_by_name('final_result:0')
40
41       predictions = sess.run(softmax_tensor, \
42                {'DecodeJpeg/contents:0': image_data})
43
44       # Sort to show labels of first prediction in order of confidence
45       top_k = predictions[0].argsort()[-len(predictions[0]):][::-1]
46       d = {'robbery':0.0, 'guns':0.0, 'vandalism':0.0, 'fist fight':0.0, 'knife'
           :0.0, 'normal':0.0}
47       count = 0
48       for node_id in top_k:
49           count += 1
50           human_string = label_lines[node_id]
51           score = predictions[0][node_id]
52           d[human_string] += score
53           # print('%s (score = %.5f)' % (human_string, score))
54       Scores = []
55       for k in d.keys():
56           Scores.append((d[k]*100, k))
57           # print(k, d[k]*100)
58       totalSum = 0
```

```
        for x in Scores:
            totalSum += x[0]

        finalList = []
        for x in range(len(Scores)):
            finalList.append((Scores[x][0]/totalSum*100, Scores[x][1]))
        finalList = sorted(finalList, reverse=True)
        for x in finalList:
            if x[1] == 'guns':
              res[x[1]].append(x[0])
            elif x[1] == 'knife':
              res[x[1]].append(x[0])
            elif x[1] == 'fist fight':
              res[x[1]].append(x[0])

# print("Average of Guns : " ,str((sum(res['guns'])/len(res['guns']))))
# print("Average of Knife : " ,str((sum(res['knife'])/len(res['knife']))))
# print("Average of fist fight : " ,str((sum(res['fist fight'])/len(res['fist
    fight']))))

avg = {
  'guns' : sum(res['guns'])/len(res['guns']),
  'knife' : sum(res['knife'])/len(res['knife']),
  'fist fight' : sum(res['fist fight'])/len(res['fist fight'])
}

print(json.dumps(avg))
```

6.3 Violence Detection Engine

Firstly we give input of video which is non-violent and the system gives maximum accuracy and labels the data as non-violent

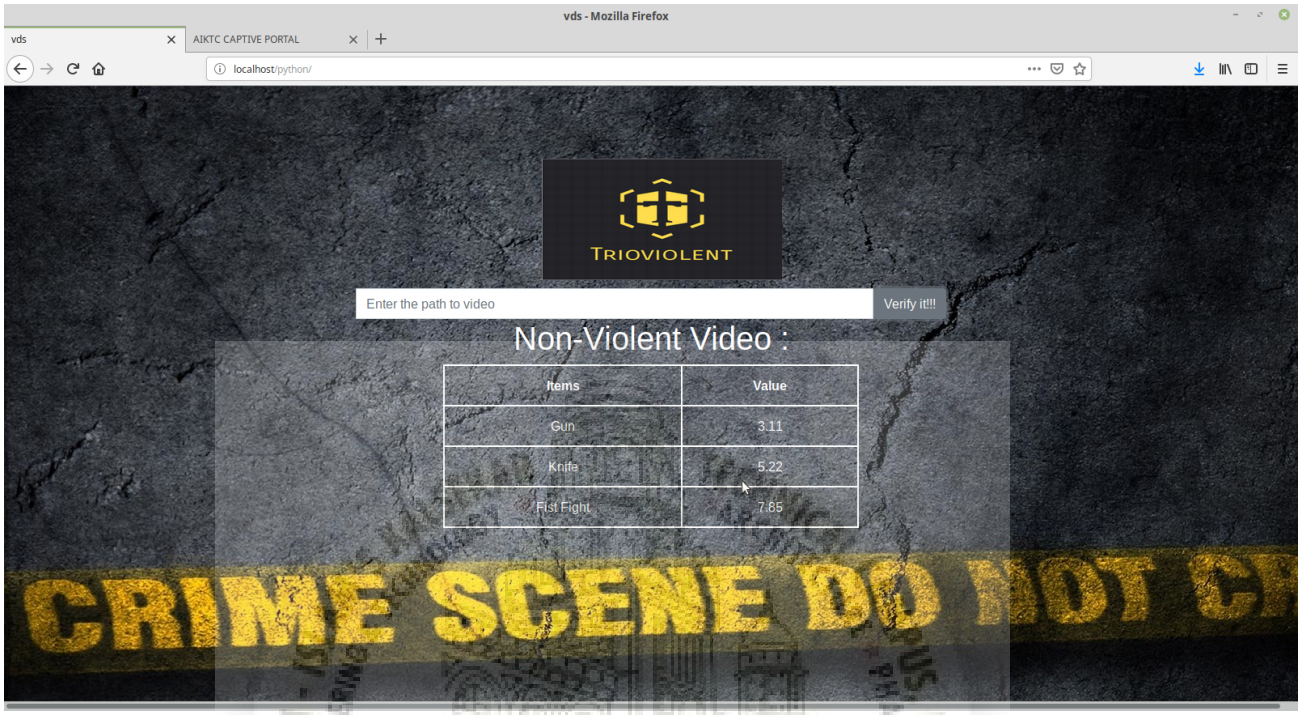


Figure 6.4: Non-Violent

Here we are taking a video as input and passing for detection and according to the detection module it is classified as violent video with respect to gun object

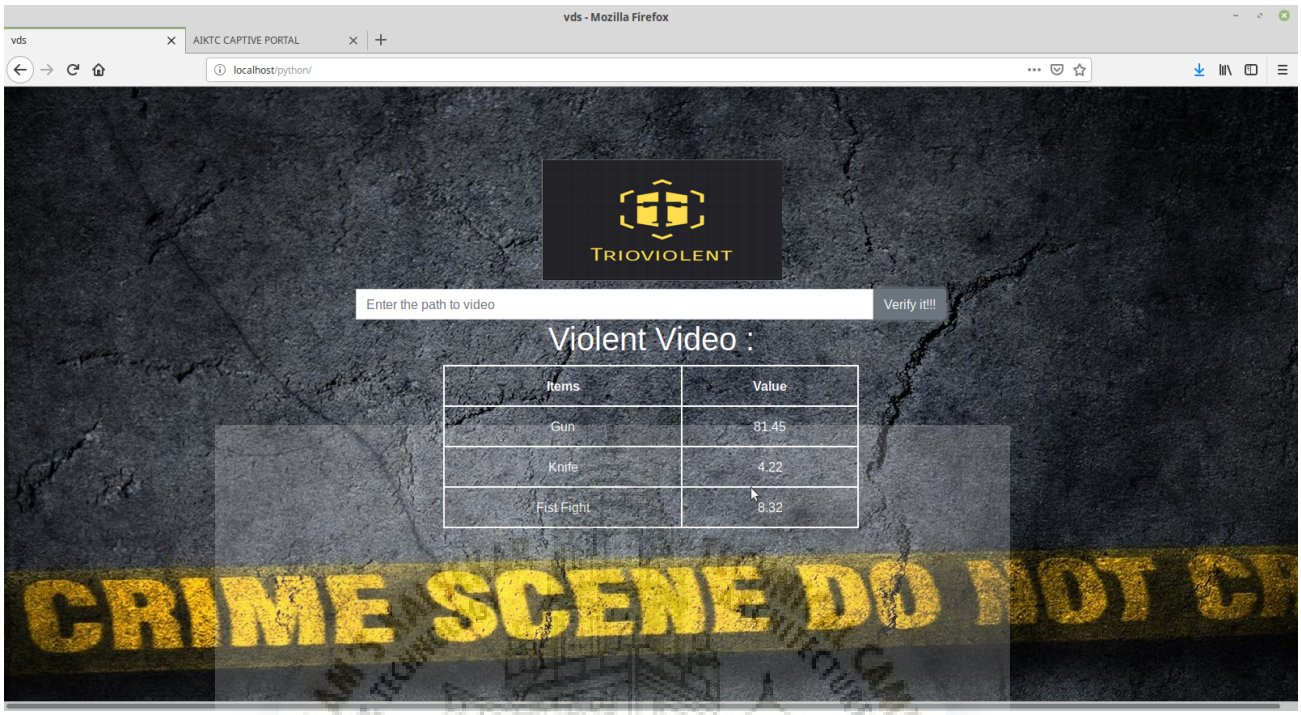


Figure 6.5: Violent

```
import cv2
import tensorflow as tf
import sys
import os
import csv
import json

vidcap = cv2.VideoCapture('test/Gun.mp4')
success,image = vidcap.read()
counts = 0

while success:
  vidcap.set(cv2.CAP_PROP_POS_MSEC,(counts*1000))
  success,image = vidcap.read()
  cv2.imwrite("test/frames/frame%d.jpg" % counts, image)
  counts += 1

# Loads label file, strips off carriage return
label_lines = [line.rstrip() for line
                  in tf.gfile.GFile("/home/trioviolent/Desktop/vds/KerasBuild/
                     tf_files/retrained_labels.txt")]

res = { 'guns' : [] , 'fist fight' : [] , 'knife' : [] }

for i in range(0,counts-1) :
  # change this as you see fit
  image_path = "test/frames/frame"+str(i)+".jpg"
  # Read in the image_data
  image_data = tf.gfile.FastGFile(image_path, 'rb').read()

  # Unpersists graph from file
```

```
  with tf.gfile.FastGFile("/home/trioviolent/Desktop/vds/KerasBuild/tf_files/
      retrained_graph.pb", 'rb') as f:
      graph_def = tf.GraphDef()
      graph_def.ParseFromString(f.read())
      _ = tf.import_graph_def(graph_def, name='')

  with tf.Session() as sess:
      # Feed the image_data as input to the graph and get first prediction
      softmax_tensor = sess.graph.get_tensor_by_name('final_result:0')

      predictions = sess.run(softmax_tensor, \
              {'DecodeJpeg/contents:0': image_data})

      # Sort to show labels of first prediction in order of confidence
      top_k = predictions[0].argsort()[-len(predictions[0]):][::-1]
      d = {'robbery':0.0, 'guns':0.0, 'vandalism':0.0, 'fist fight':0.0, 'knife'
          :0.0, 'normal':0.0}
      count = 0
      for node_id in top_k:
          count += 1
          human_string = label_lines[node_id]
          score = predictions[0][node_id]
          d[human_string] += score
          # print('%s (score = %.5f)' % (human_string, score))
      Scores = []
      for k in d.keys():
          Scores.append((d[k]*100, k))
          # print(k, d[k]*100)
      totalSum = 0
      for x in Scores:
          totalSum += x[0]

      finalList = []
      for x in range(len(Scores)):
          finalList.append((Scores[x][0]/totalSum*100, Scores[x][1]))
      finalList = sorted(finalList, reverse=True)
      for x in finalList:
          if x[1] == 'guns':
            res[x[1]].append(x[0])
          elif x[1] == 'knife':
            res[x[1]].append(x[0])
          elif x[1] == 'fist fight':
            res[x[1]].append(x[0])

# print("Average of Guns : " ,str((sum(res['guns'])/len(res['guns']))))
# print("Average of Knife : " ,str((sum(res['knife'])/len(res['knife']))))
# print("Average of fist fight : " ,str((sum(res['fist fight'])/len(res['fist
    fight']))))

avg = {
  'guns' : sum(res['guns'])/len(res['guns']),
  'knife' : sum(res['knife'])/len(res['knife']),
  'fist fight' : sum(res['fist fight'])/len(res['fist fight'])
}

print(json.dumps(avg))
```

Chapter 7

System Testing

Everything which is developed should get tested. Because if developed software has some errors that may cost users business. System testing is the testing in which fully integrated software are tested. Basically system testing is process of checking if developed software is working as per users requirements it fully observed by computer based system.Testing is important because in Software Development Life Cycle the system is perform as the first level of testing where system is tested as a whole .During testing validation and verification both are require.

7.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Object Detection	Object Detection	Results after object detection	Successful
T02	Recognition	Action Recognition	classify violent and non-violent behavior v	Successful

7.2 Sample of a Test Case

Title: Detection
Description: System will detect objects which are violent and show it to the user with respect to average count depending on its violent and non-violent nature

Precondition: System must satisfy all the perquisites to run object detection, like it must contain all the object detection libraries provide by OpenCV
Assumption: A supported browser is being used.

Test Steps:

1. Navigate to V.D.S Homepage
2. In the search area browse any video file .
3. Click on 'Verify' button.
4. After clicking on verify button in the background detection code will start its work and will provide us with the desired output

Expected Result: A page displaying the average count of various objects and actions with the classification of whether the video is violent or not

Actual Result: Successfully classified as violent or not
upload the image of result

7.2.1 Software Quality Attributes

1. Availability: The system will be available 24/7 as application is totally based on web ,whenever the user use the system the specific data should be available to the user.

2. Correctness: The system must predict the object and action correctly with respect to the video provided

3. Reliability: The system should be reliable for producing correct output so that user can reliable on system.

4. Extensibility: The system is capable to be modified by changing some modules or by adding some features to the existing system.

Chapter 8

Screenshots of Project

8.1 Log-in Page

Figure 8.1: Log-In Page

8.2 Classification of Violence

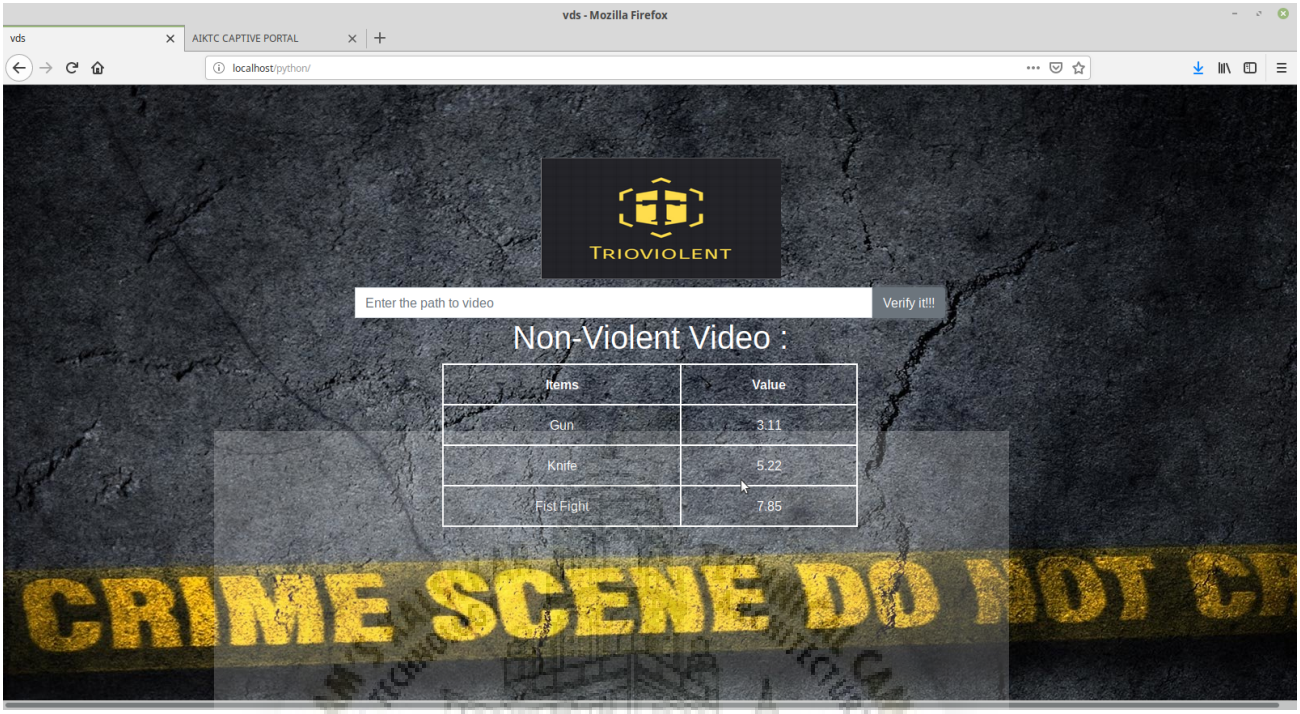


Figure 8.2: Non-Violent

Figure 8.3: Frames of Non-Violent video

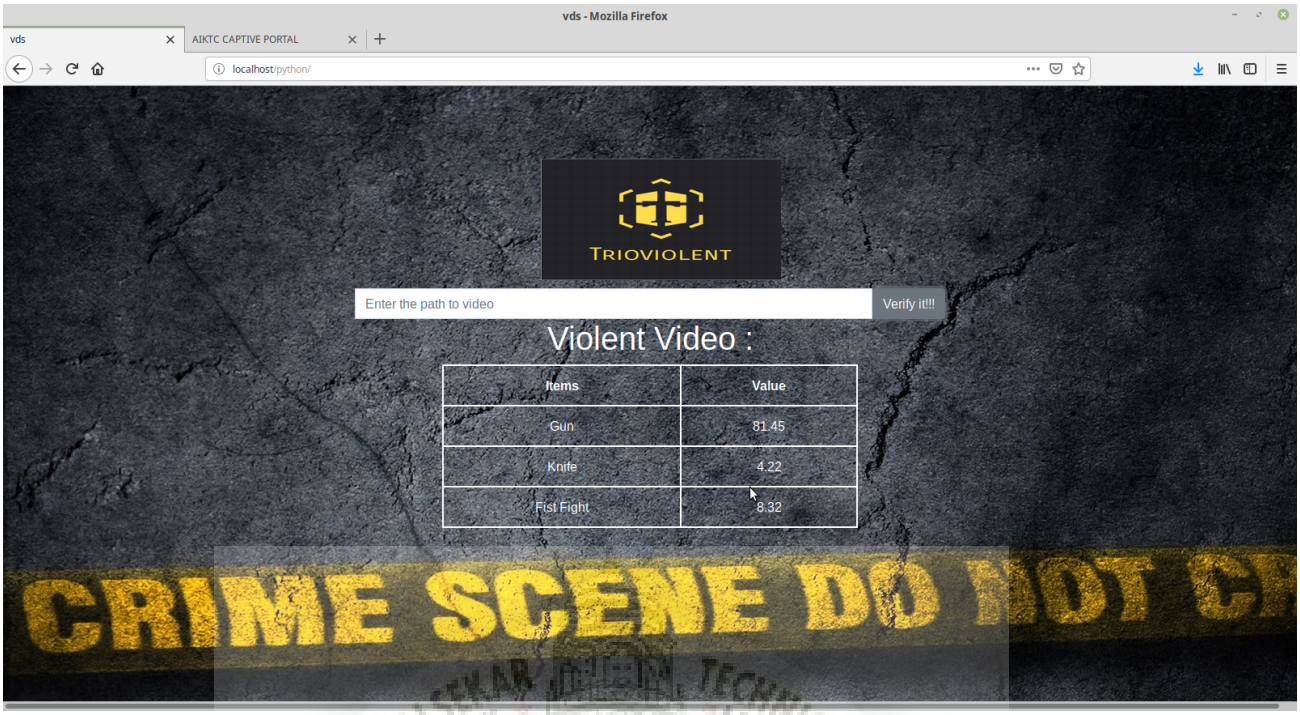


Figure 8.4: Violent with respect to gun

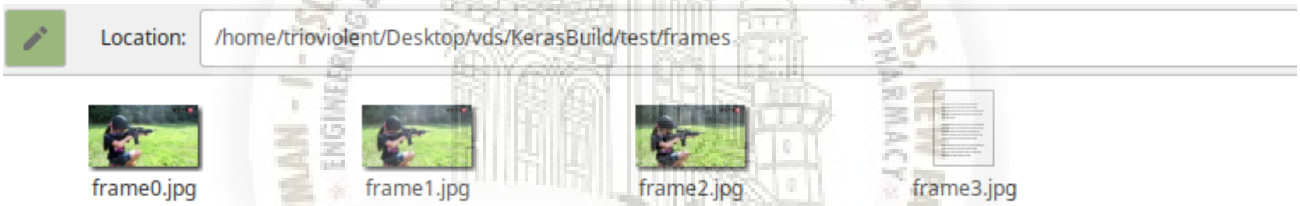


Figure 8.5: Frames of Violent with respect to gun

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

- The proposed system will be able to classify feed of CCTV or any other video into violent and non-violent category.
- The system will also have the privilege to store the violent outbreak situations in the form of frames, and to notify about the same to the official authorities to reduce the headache that comes with these kind of scenes if it is done manually

9.2 Future Plan

- As in our system we are trying to work in classifying the video is violent or not between two people.
- To take a bigger crowd into surveillance for violent detection and classification of their violent behaviour
- It will work on live feed and take that feed as an input to classify into violent or non-violent

References

[1] *Automatic fight detection in surveillance videos*; Fu, Eugene Yujun and Leong, Hong Va and Ngai, Grace and Chan, Stephen CF, International Journal of Pervasive Computing and Communications, 2017

[2] *Temporal segment networks: Towards good practices for deep action recognition*; Wang, Limin and Xiong, Yuanjun and Wang, Zhe and Qiao, Yu and Lin, Dahua and Tang, Xiaoou and Van Gool, Luc, European Conference on Computer Vision, Springer, 2016

[3] *Evaluation of multiple features for violent scenes detection*; Lam, Vu and Phan, Sang and Le, Duy-Dinh and Duong, Duc Anh and Satoh, Shin'ichi, Multimedia Tools and Applications , Springer, 2017

[4] *Video-based abnormal human behavior recognition—A review*; Popoola, Oluwatoyin P and Wang, Kejun, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), IEEE, 2012

[5] *Effective design, configuration, and use of digital cctv*; Keval, Hina Uttam, UCL (University College London), 2009

Achievements

1. Conferences

   Violence Detection System using DSP processor; Sayed Najneen Fatma Mustaq Ali Parveen Fatima, Chougle Zaid Noornabi Zerin, Kothari Hozefa Abbas Jumana, Conference on Technologies for Future Cities-2019 ,(Venue : Pillai College of Engineering)

2. Project Competitions

   Violence Detection System using DSP processor; Sayed Najneen Fatma Mustaq Ali Parveen Fatima, Chougle Zaid Noornabi Zerin, Kothari Hozefa Abbas Jumana, Avishkar Research Project ,December-2018 (Venue :RAIT)

University of Mumbai

DEPARTMENT OF
STUDENTS' DEVELOPMENT

13th Inter-Collegiate/
Institute/Department

Avishkar

Research Convention: 2018-19

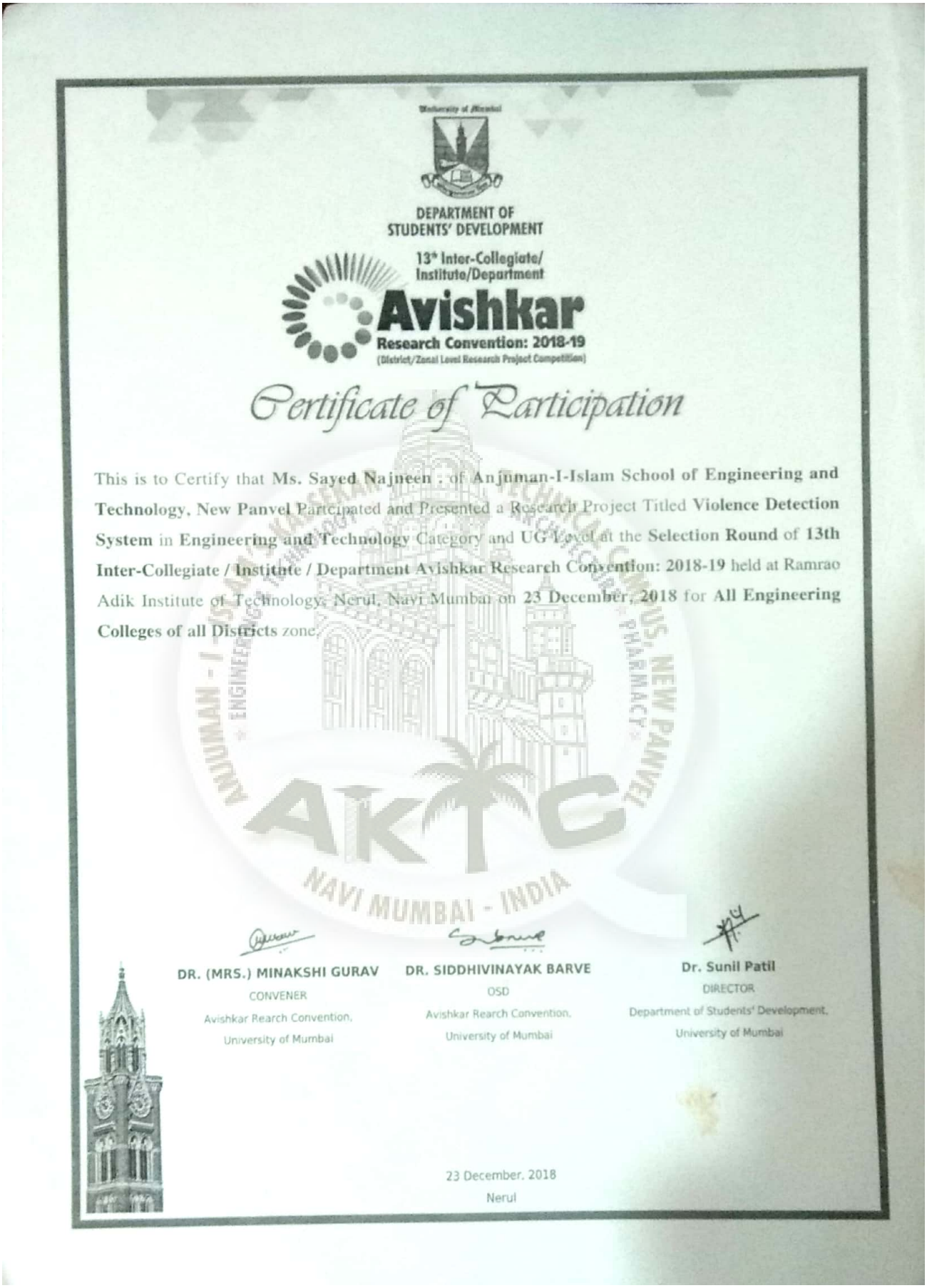
(District/Zonal Level Research Project Competition)

Certificate of Participation

This is to Certify that **Mr. Chougle Zaid Noornabi** of **Anjuman-I-Islam School of Engineering and Technology, New Panvel** Partcipated and Presented a Research Project Titled **Violence Detection System** in **Engineering and Technology** Category and **UG Level** at the **Selection Round** of **13th Inter-Collegiate / Institute / Department Avishkar Research Convention: 2018-19** held at Ramrao Adik Institute of Technology, Nerul, Navi Mumbai on **23 December, 2018** for **All Engineering Colleges of all Districts** zone.

DR. (MRS.) MINAKSHI GURAV	DR. SIDDHIVINAYAK BARVE	Dr. Sunil Patil
CONVENER	OSD	DIRECTOR
Avishkar Rearch Convention,	Avishkar Rearch Convention,	Department of Students' Development,
University of Mumbai	University of Mumbai	University of Mumbai

23 December, 2018
Nerul



DEPARTMENT OF
STUDENTS' DEVELOPMENT

13th Inter-Collegiate/
Institute/Department

Avishkar

Research Convention: 2018-19

(District/Zonal Level Research Project Competition)

Certificate of Participation

This is to Certify that **Ms. Sayed Najneen** of **Anjuman-I-Islam School of Engineering and Technology, New Panvel** Participated and Presented a Research Project Titled **Violence Detection System** in **Engineering and Technology** Category and **UG** Level at the **Selection Round** of **13th Inter-Collegiate / Institute / Department Avishkar Research Convention: 2018-19** held at Ramrao Adik Institute of Technology, Nerul, Navi Mumbai on **23 December, 2018** for **All Engineering Colleges of all Districts** zone.

DR. (MRS.) MINAKSHI GURAV
CONVENER
Avishkar Rearch Convention,
University of Mumbai

DR. SIDDHIVINAYAK BARVE
OSD
Avishkar Rearch Convention,
University of Mumbai

Dr. Sunil Patil
DIRECTOR
Department of Students' Development,
University of Mumbai

23 December, 2018
Nerul

University of Mumbai

DEPARTMENT OF
STUDENTS' DEVELOPMENT

13th Inter-Collegiate/
Institute/Department

Avishkar

Research Convention: 2018-19

(District/Zonal Level Research Project Competition)

Certificate of Participation

This is to Certify that **Mr. Kothari Hozefa Abbas** of **Anjuman-I-Islam School of Engineering and Technology, New Panvel** Partcipated and Presented a Research Project Titled **Violence Detection System** in **Engineering and Technology** Category and **UG** Level at the **Selection Round** of **13th Inter-Collegiate / Institute / Department Avishkar Research Convention: 2018-19** held at Ramrao Adik Institute of Technology, Nerul, Navi Mumbai on **23 December, 2018** for **All Engineering Colleges of all Districts** zone.

DR. (MRS.) MINAKSHI GURAV
CONVENER
Avishkar Rearch Convention,
University of Mumbai

DR. SIDDHIVINAYAK BARVE
OSD
Avishkar Rearch Convention,
University of Mumbai

Dr. Sunil Patil
DIRECTOR
Department of Students' Development,
University of Mumbai

23 December, 2018
Nerul

Mahatma Education Society's

Pillai College of Engineering

PCE
PILLAI COLLEGE OF ENGINEERING

CTFC 2019

CONFERENCE ON
TECHNOLOGIES FOR FUTURE CITIES
8TH - 9TH JAN 2019

Certificate of Participation

This is to certify that CHOUGLE ZAID NOOR"ABI has Presented a paper titled Violence Detection System using Digital Signal Processor in the track Hardware Technologies of Conference on "Technologies for Future Cities 2019", held during 8th - 9th Jan 2019.

Dr. S M Joshi
Convener

Dr. P S Goyal
Secretary

Mahatma Education Society's

Pillai College of Engineering

CONFERENCE ON

TECHNOLOGIES FOR FUTURE CITIES

8TH - 9TH JAN 2019

Certificate of Participation

This is to certify that Sayed Najneen Fatma has Presented a paper titled Violence Detection System Using Digital Signal Processor in the track Hardware Technologies of Conference on "Technologies for Future Cities 2019", held during 8th - 9th Jan 2019.

Dr. S M Joshi
Convener

Dr. P S Goyal
Secretary

Mahatma Education Society's

Pillai College of Engineering

CONFERENCE ON

TECHNOLOGIES FOR FUTURE CITIES

8TH - 9TH JAN 2019

Certificate of Participation

This is to certify that KOTHARI HUZEFA ABBAS has Presented a paper titled Violence Detection System Using Digital Signal Processor in the track Hardware Technologies of Conference on "Technologies for Future Cities 2019", held during 8th - 9th Jan 2019.

Dr. S M Joshi
Convener

Dr. P S Goyal
Secretary

Violence Detection System using Digital Signal Processor

1st Sayed Najneen Fatma Mustaq Ali Parveen Fatima
dept. of computer engineering
Anjuman-I-Islam's Kalsekar Technical Campus
New Panvel,Navi Mumbai, India
nazof1597@gmail.com

2nd Chougle Zaid Noornabi Zerin
dept. of computer engineering
Anjuman-I-Islam's Kalsekar Technical Campus
New Panvel,Navi Mumbai, India
zaid295@gmail.com

3rd Kothari Hozefa Abbas Jumana
dept. of computer engineering
Anjuman-I-Islam's Kalsekar Technical Campus
New Panvel,Navi Mumbai, India
kotharihozefa53@gmail.com

***Abstract*—The age of data is here and looking dead into our eyes with some promising data, with this data a field of computational science i.e. machine learning is fancying its chances of existence. Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms needs lots of data and data age is providing the same, because of that techniques like action recognition, gesture recognition and object detection are coming into play and can be used for various day-to-day tasks both in commercial and private sectors**
On that note our system will recognize the violent behavior incoming from the CCTV feed and it will be working on real-time data. As the processing will be done in real time it's speed should be up to the mark so to achieve that kind of optimal and high success rate of data processing will be done using DSP processor TMS320Cx family which is generally used for image and video processing so we will be using TMS320C family DSP processor for detection of violent behavior incoming from CCTV feed and segregate the feed in violent or non-violent behavior

***Index Terms*—Violence, Non-violence, DSP-kit, Optical flow, Hidden Markov Model (HMM), Temporal Structural Network (TSN), Support Vector Machines (SVM), Action Similarity Labelling Challenge (ASLAN), Violent Flow Descriptors(ViF), videos, image processing, Region of Convergence Curve(ROC), Standard Deviation(SD), Digital Signal Controller(DSC).**

I. Introduction

Our project promises to replace the ongoing manual work of violence detection with an automated approach using deep learning techniques. It will detect the violence from the incoming CCTV feed [1], [2] and will provide us with the output whether it is violent or not, we will categorize whether the video is violent or not on more then one parameter i.e, if it is performing a certain kind of action [3] or using a specific object. [4] We will be doing both the action recognition and object detection for more accurate classification of violent behavior .

A. Motivation

It will reduce the mayhem that is caused due to a missed frame at the time of surveillance, or a missed frame because of some human behavior It will automate the surveillance system to detect whether a single video is violent or not so that the cost to the society both in money and blood can be minimized.

B. Objective and Scope

The main objective of our system is to :

- To identify the violent behaviour.
- To store detailed information of the violent outbreak to ease up the investigation.

The scope of our project is as follows :

- Our system will focus on one-to-one person.
- It will detect weapons like guns and knifes.
- Notification will be given of violent behaviour, if found in any locality.
- A beep sound will be produced whenever there is pop-up of violent frame.

II. Methodology

In our system we simply are going classify a video into violent or non-violent class but as simple it sounds it isn't. For a human to recognize whether a particular scenario, movie scene or a image is violent or not as because our mind is trained in that fashion in our day-to-day life we have seen many scenarios many images which depict certain kind of violent behavior so we i.e our brain can categorize violence from a normal scene. We are trying to develop such a system that can adapt what our brain uses for classifying violence from a normal scenario. Our system is solely based on neural networks, we will be taking in video from CCTV video [5] and we will be classifying the incoming feed in violent or not. As the processing of the data will be done in real time the processing speed should be up to the mark. To attain that

real time processing speed our processing of data will be done by Digital signal processor of family TMS320Cx [6] [7] as this processor is mainly used for handling time data. After the video is divided in violent or not our system will notify the authorities about the violent outbreak

A. Use-case

Firstly , User will Register or log in . After that, user will be authenticate ,for this we will use SHA-256 Algorithm. As the user is authorized he/she will be able to see the real time video which is going on and a notification will arrive if any violent behaviour is detected. If the user is not authenticated then he/she have to log in again.

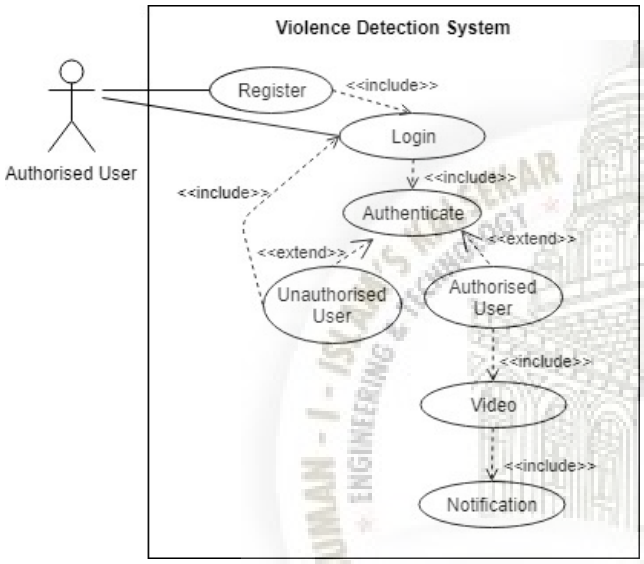


Fig. 1. Use Case

III. OVERVIEW OF THE SYSTEM

The real time video will be taken as an input which will be process through Digital Signal Processor. After the video is processed it will go to Extraction Engine module. Here, the processed video is divided into frames and those are sent for denosing [8] i.e any extra material or disturbance in the frames of images will be removed

After denoising of frames of images is done it will move towards Violence Detection Engine. In this engine we are extracting specific features [9]from the frames and we are comparing those features with our data-set to check whether the extracted features matches to our category of violent behavior, if it matches then it will be stored in the database and the necessary authorities will be alerted, if not then next frame will be taken in for detection

A. Extraction Engine

- First to convert the video into frames of images we will be using OpenCV which is python library.

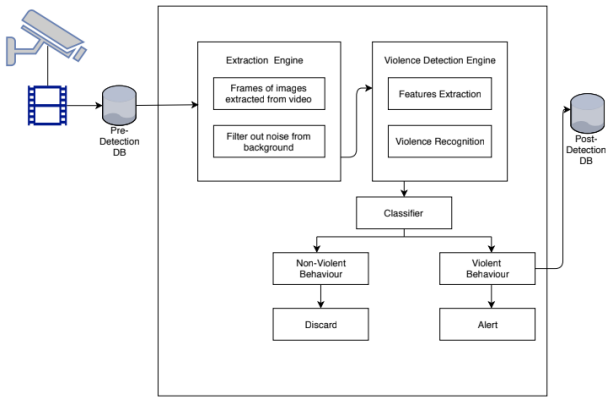


Fig. 2. System Architecture

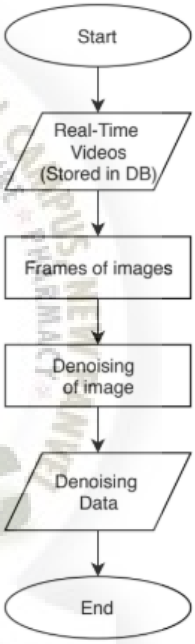


Fig. 3. Extraction Engine.

- Later on Denoising of the frames is done using OpenCV library.
- OpenCV(Open Source Computer Vision Library) is designed for computational efficiency and with a strong focus on real-time applications.
- The library can take advantage of multi-core processing.

B. Violence Detection Engine

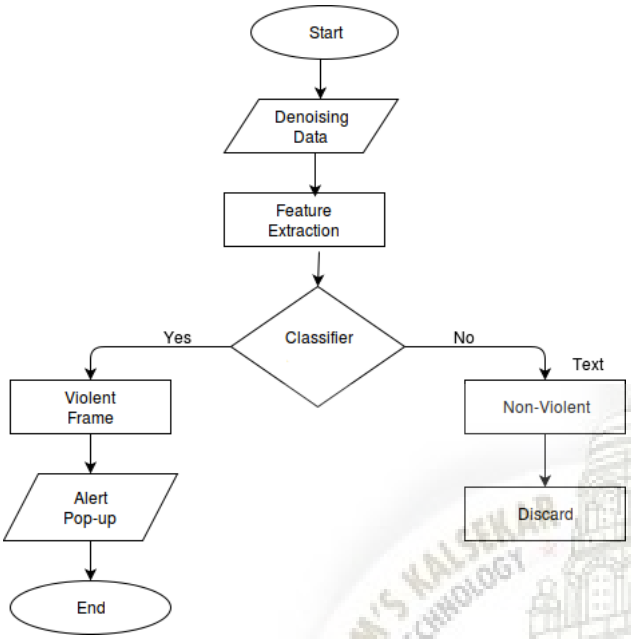


Fig. 4. Violence Detection Engine.

In this module denoised image is taken in as input and the intended features are searched and extracted from the image

After successfully extracting the features it is sent into the classifier, here will be using either Support Vector Machines(SVM) or Temporal Segment Networks(TSN) [10]depending on various parameters like time taken for processing, accuracy rate, etc

After the classification if any violent tagged frames are found in the frames then those frames are meant to be stored in the database

TSN(temporal segment networks framework) is a framework for video-based human action recognition.

TSN effectively models long-range temporal dynamics by learning from multiple segments of one video in an end-to-end manner.

SVM(support vector machines) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

Conclusion

The proposed system will be able to classify incoming feed of CCTV into violent and non-violent system, it will reduce the load of an observer who is assigned to monitor the CCTV feeds The system will also have the privilege to store the violent outbreak situations in the form of frames, and to notify about the same to the official authorities to reduce the headache that comes with these kind of scenes if it is done manually

References

[1] O. P. Popoola and K. Wang, "Video-based abnormal human behavior recognitiona review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 865–878, 2012.

[2] H. U. Keval, *Effective design, configuration, and use of digital cctv*. PhD thesis, UCL (University College London), 2009.

[3] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–I, IEEE, 2001.

[5] A. Anema, S. Kluberg, K. Wilson, R. S. Hogg, K. Khan, S. I. Hay, A. J. Tatem, and J. S. Brownstein, "Digital surveillance for enhanced detection and response to outbreaks," *The Lancet Infectious Diseases*, vol. 14, no. 11, pp. 1035–1037, 2014.

[6] T. Instruments, "Spru615, tms320dm642 technical overview: Dsp video and imaging digital applications." http://www.ti.com/lit/ug/spru615/spru615.pdf. Accessed on Jan 2012.

[7] T. Instruments, "Sprui98i, tms320c6000 programmer's guide." http://www-s.ti.com/sc/techlit/SPRU198. Accessed on Jan 2012.

[8] O. Documents. http://docs.opencv.org/. Accessed on April 2012.

[9] V. Lam, S. Phan, D.-D. Le, D. A. Duong, and S. Satoh, "Evaluation of multiple features for violent scenes detection," *Multimedia Tools and Applications*, vol. 76, no. 5, pp. 7041–7065, 2017.

[10] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European Conference on Computer Vision*, pp. 20–36, Springer, 2016.

Appendix

TMS320F28335 Digital Signal Controller (DSC)

Features :

itemsep=1 mm, parsep=0pt

- High-Performance Static CMOS Technology
  - Up to 150 MHz (6.67-ns Cycle Time)
  - 1.9-V/1.8-V Core, 3.3-V I/O Design
- High-Performance 32-Bit CPU (TMS320C28x)
  - IEEE 754 Single-Precision Floating-Point Unit (FPU) (F2833x Only)
  - 16 16 and 32 32 MAC Operations
  - 16 16 Dual MAC
  - Harvard Bus Architecture
  - Fast Interrupt Response and Processing
  - Unified Memory Programming Model
  - Code-Efficient (in C/C++ and Assembly)
- Six-Channel DMA Controller (for ADC, McBSP, ePWM, XINTF, and SARAM)
- 16-Bit or 32-Bit External Interface (XINTF)
  - More Than 2M 16 Address Reach
- On-Chip Memory
  - F28335, F28333, F28235: 256K 16 Flash, 34K 16 SARAM
  - F28334, F28234: 128K 16 Flash, 34K 16 SARAM
  - F28332, F28232: 64K 16 Flash, 26K 16 SARAM
  - 1K 16 OTP ROM
- Boot ROM (8K 16)

  - With Software Boot Modes (Through SCI, SPI, CAN, I2C, McBSP, XINTF, and Parallel I/O)
  - Standard Math Tables
- Clock and System Control
  - On-Chip Oscillator
  - Watchdog Timer Module
- GPIO0 to GPIO63 Pins Can Be Connected to One of the Eight External Core Interrupts
- Peripheral Interrupt Expansion (PIE) Block That Supports All 58 Peripheral Interrupts
- 128-Bit Security Key/Lock
  - Protects Flash/OTP/RAM Blocks
  - Prevents Firmware Reverse Engineering
- Enhanced Control Peripherals
  - Up to 18 PWM Outputs
  - Up to 6 HRPWM Outputs With 150 ps MEP Resolution
  - Up to 6 Event Capture Inputs
  - Up to 2 Quadrature Encoder Interfaces
  - Up to 8 32-Bit Timers (6 for eCAPs and 2 for eQEPs)
  - Up to 9 16-Bit Timers (6 for ePWMs and 3 XINTCTRs)
- Three 32-Bit CPU Timers
- Serial Port Peripherals
  - Up to 2 CAN Modules
  - Up to 3 SCI (UART) Modules
  - Up to 2 McBSP Modules (Configurable as SPI)
  - One SPI Module
  - One Inter-Integrated Circuit (I2C) Bus
- 12-Bit ADC, 16 Channels
  - 80-ns Conversion Rate
  - 2 8 Channel Input Multiplexer
  - Two Sample-and-Hold
  - Single/Simultaneous Conversions
  - Internal or External Reference
- Up to 88 Individually Programmable, Multiplexed GPIO Pins With Input Filtering
- JTAG Boundary Scan Support
  - IEEE Standard 1149.1-1990 Standard Test Access Port and Boundary Scan Architecture
- Advanced Emulation Features
  - Analysis and Breakpoint Functions
  - Real-Time Debug Using Hardware
- Development Support Includes
  - ANSI C/C++ Compiler/Assembler/Linker
  - Code Composer Studio IDE
  - DSP/BIOS and SYS/BIOS
  - Digital Motor Control and Digital Power Software Libraries
- Low-Power Modes and Power Savings
  - IDLE, STANDBY, HALT Modes Supported
  - Disable Individual Peripheral Clocks
- Endianness: Little Endian

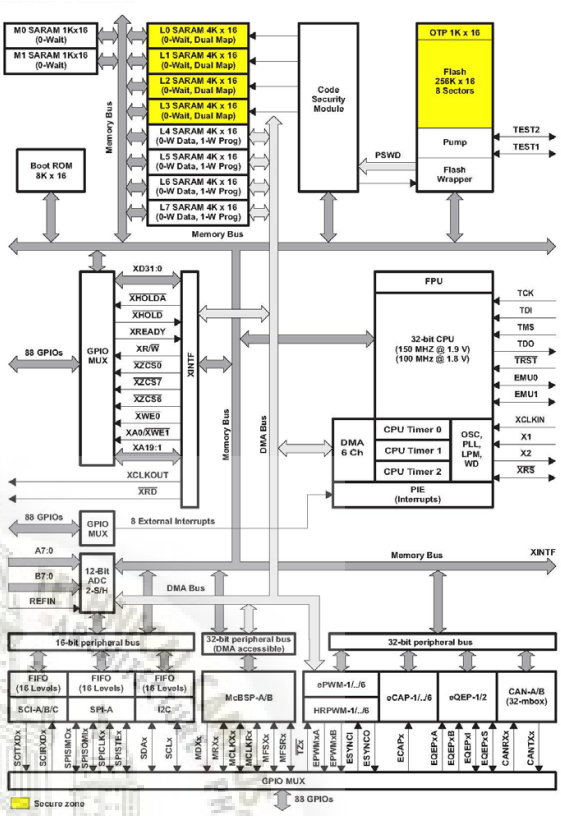


Fig. 5. Functional Block Diagram

- Package Options
  - Lead-free, Green Packaging
  - Plastic Ball Grid Array (BGA) (ZJZ)
  - MicroStar BGA (ZHH)
  - Low-Profile Quad Flatpack (LQFP) (PGF)
  - Thermally Enhanced Low-Profile Quad Flatpack (HLQFP) (PTP)
- Temperature Options
  - A: 40C to 85C (PGF, ZHH, ZJZ)
  - S: 40C to 125C (PTP, ZJZ)
  - Q: 40C to 125C (PTP, ZJZ) (AEC Q100 Qualification for Automotive Applications)

Applications :

- Industrial AC Inverter Drives
- Industrial Servo Amplifiers and Controllers
- Computer Numerical Control (CNC) Machining
- Uninterruptible and Server Power Supplies
- Telecom Equipment Power
- Solar Inverters