

A Mini Project report on
CRIME DETECTOR USING MACHINE LAERNING

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE ENGINEERING - CYBER SECURITY(CSE-CS)

Submitted By

Y.SAI KIRAN (21EG109B06)
G.SACHIN KUMAR (21EG109B19)
M.SAITEJA (22EG509B04)

Under the guidance of
Mrs.N.Nagalakshmi
Assistant Professor



Department of Information Technology

ANURAG UNIVERSITY

(Approved by AICTE and NBA Accredited) Venkatapur (V),

Ghatkesar (M), Medchal district, Hyderabad,

Telangana,500088

2021-2025

ANURAG UNIVERSITY
Venkatapur (V), Ghatkesar (M), Medchal District,
Hyderabad, Telangana,500088
2020-2024
Department of Information Technology



CERTIFICATE

This is to certify that the project report entitled “**CRIME DETECTOR USING MACHINE LAERNING**” is a Bonafide work done and submitted by **Y.Saikiran-(21EG109B06),G.sachin kumar-(21EG109B19),M.saiteja-(22EG509b04)** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science Engineering (CSE-CS)** from Anurag University, Hyderabad during the academic year 2021- 2025. The Bonafide work has not been submitted elsewhere for the award of any other degree.

Signature of Supervisor

Mrs.N.Nagalakshmi
Assistant Professor
Department of IT

Signature of HOD

K. Sudeer Reddy
HOD,IT

External Examiner

ACKNOWLEDGEMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **N.nagalaxkshmi, Assistant Professor, Department of Information Technology**, Anurag University for his constant encouragement and inspiring guidance without which this project could not have been completed. His critical reviews and constructive comments improved our grasp of the subject and steered us towards the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

We would like acknowledge our sincere gratitude for the support extended by **Dr. Anand Babu, Professor, Department of Information Technology**, Anurag University. We also express my deep sense of gratitude to **K. Sudeer Reddy, HOD**, and Project review committee members, whose research expertise and commitment to the highest standards continuously motivated me during the crucial stage our project work.

We would like express our special thanks to **Dr. V. VIJAYA KUMAR, Dean School of Engineering**, Anurag University, for his encouragement and timely support in our B. Tech program.

Y. SAIKIRAN (21EG109B06)

G. SACHINKUMAR(21EG109B19)

M. SAI TEJA (22EG509B04)

DECLARATION

This is to Certify that the project work entitled “**CRIME DETECTOR USING MACHINE LAERNING** submitted to JNTUH in partial fulfillment of the requirement for the award of the Degree of Bachelor of Technology (B-Tech), is an original work carried out by **Y.Saikiran (21EG109B06), G.Sachin kumar(21EG109B19),M.Saiteja(22EG509B04)**under the guidance of **Mrs.N.Nagalakshmi Designation** in the Department of Information Technology. This matter embodied in this project is a genuine work, done by the students and has not been submitted whether the university or to any other university/Institute for the fulfillment of the requirement of any course of study.

Y. SAIKIRAN (21EG109B06)

G. SACHINKUMAR(21EG109B19)

M. SAI TEJA (22EG509B04)

+ CONTENTS

CHAPTERS	PAGE
List of Figures	i
List of Abbreviations	ii
List of Tables	ii
1. Introduction	
1.1 Overview	1
1.2 Objective	1
1.3 Problem Formulation	2
1.4 Scope of the Project	2
1.5 Feasibility	3
1.6 System Requirements	
1.6.1 Software Requirements	6
1.6.2 Hardware Requirements	6
2. Requirements	
2.1 Requirements Specification	7
2.2 User interface	9
3. Analysis	
3.1 Use Case Model	10
3.1.1 Use Case Diagram	
3.1.2 Activity Diagram	10
4. Design	
4.1 UML diagram	11
4.2 Sequence diagram	12
4.3 Class diagram	13
4.4 Collaboration diagram	13
5. Implementation	14
6. User Screens	21
7. Testing Methodology	32
8. Conclusions and Future Enhancements	35
9. References	35

LIST OF FIGURES

S.no	Figure number	Description	Page number
1	4.1.1	Machine Learning Algorithm Process Architecture	6
2	4.3.1	Use Case Diagram	9
3	4.3.2	Class Diagram	10
4	4.3.3	Sequence Diagram	11
5	4.3.4	Collaboration Diagram	12
6	4.3.5	Activity Diagram	13
7	4.3.6	Deployment Diagram	14
8	5.1,5.2	Datasets	16
9	7.2.1 ,	Datasets	17
10	7.2.2	Phishing URLs	28
11	7.2.3	Starting Django Server	29
12	9.1	Admin Login Page	30
13	9.2	Run SVM Algorithm	30
14	9.3	Run Light GBM Algorithm	30
15	9.4	Test URL	31
16	9.5	Output	31

ABSTRACT

Intruder detection in protected zones can be achieved through physical measures like patrols and CCTV monitoring, as well as active protection elements such as electronic alarm systems. These systems can be classified based on the area where they detect intrusions: point detection (e.g., object protection systems), boundary detection (e.g., perimeter protection), and area detection (e.g., area protection systems). Modern camera systems offer advanced capabilities like motion detection and event analysis, with intruder detection often requiring confirmation from at least two detectors, such as CCTV or physical security personnel. In the case of phishing attacks, the goal is to steal sensitive information like usernames, passwords, and bank details. This explores the detection of phishing URLs using machine learning techniques, including K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Support Vector Machine (SVM), and LightGBM algorithms. By analyzing various features of legitimate and phishing URLs, the performance of these algorithms is compared based on accuracy, false positives, and false negatives to identify the most effective method for phishing detection..

1.INTRODUCTION

1.1 OVERVIEW

Detecting suspicious activities in public places has become an important task due to the increasing number of shootings, knife attacks, terrorist attacks, etc. happening in public places all around the world. This project focuses on a deep learning approach to detect suspicious activities using Convolutional Neural Networks from images and videos. Currently, surveillance cameras play a vital role to ensure the safety of the people, yet they are plain video-providing entities with no smart decision-making mechanisms of their own. Because of this growth of data composed from surveillance cameras, automated video streams have become a requisite for automatically detecting abnormal events

1.2 OBJECTIVE

The objective of this project is to investigate and enhance intruder detection systems within a protected zone by utilizing both physical and active protection elements. The research will focus on improving the probability of detecting intruders through various methods, including electronic safety systems, alarm systems, and advanced CCTV camera systems. Specifically, the study will evaluate detection mechanisms such as point detection systems (object protection systems), boundary detection systems (perimeter protection systems), and area detection systems (area protection systems). Additionally, the project will examine the effectiveness of automatic intruder detection features integrated into camera systems, such as motion detection, event analysis, and security incident verification. The project aims to optimize the reliability of detection and security response by verifying alarms through multiple detectors and camera-based surveillance within the protected zone.

1.3PROBLEM FORMULATION:

The experiments were performed in compliance with the technical norms for CCTV security systems (62676-4) and with the requirements for transmissions (62676-1-2). A person walking at a normal pace served as a testing target. The experiments were carried out in a way that allows for comparison of individual results. The camera surveillanced a constantly lit environment, which did not contain any distracting elements. The test target repeatedly carried out the same movements in the same distance to the camera and fit into 100% of the camera image's height. This eliminated the possibility, that a change in camera's resolution might influence the resolution attributed to the moving object. Each test consisted of 100 repetitions.

1.4: SCOPE OF THIS :

The scope of this focuses on improving the efficiency and reliability of intruder detection systems within protected zones by analyzing both physical and electronic security measures. The project will study and compare different detection methods, including point detection (object protection systems), boundary detection (perimeter protection systems), and area detection systems, to assess their effectiveness in various contexts. It will evaluate active protection elements, such as alarm systems, electronic safety systems, and CCTV camera systems, with particular emphasis on recent advancements in camera technology, including motion detection and automatic event analysis. Additionally, the project will explore methods for verifying alarms through multiple detectors or by integrating camera systems with physical surveillance to ensure accurate alarm verification before initiating a security response. The project also aims to optimize security responses by combining physical patrols, surveillance technology, and alarm systems. Probabilistic modeling will be used to analyze and improve the likelihood of successful intruder detection in different scenarios, ultimately providing recommendations for enhancing the accuracy and efficiency of detection and response protocols in secured zones.

1.5 FEASIBILITY:

The feasibility of this is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

1.5.1:Economical feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

1.5.2. Technical feasibility

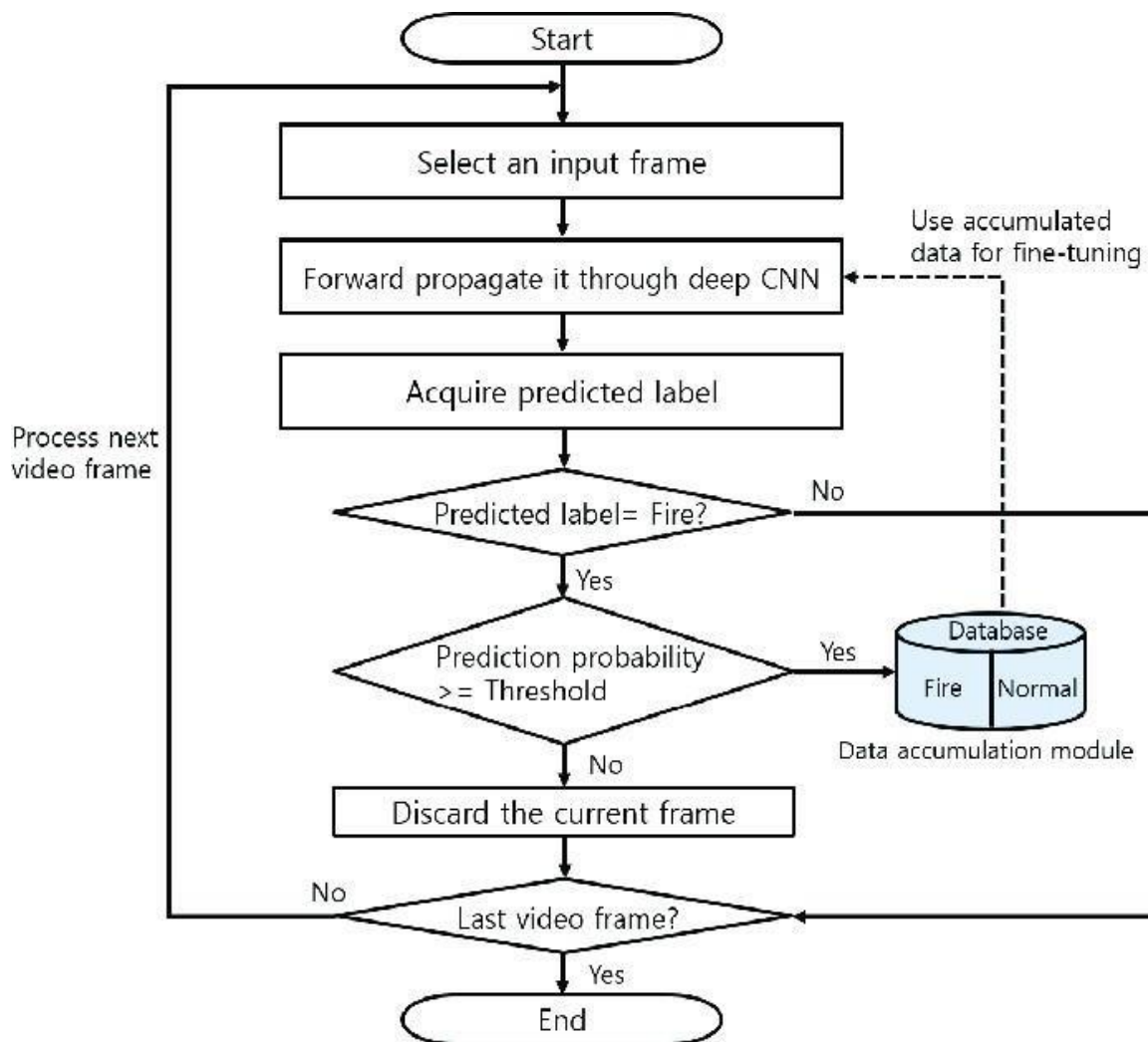
This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

.

1.5.3 Social feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed educate user about system and to make him familiar with it. His level of confidence must raised so that he is able make some constructive criticism, which is welcomed, as he is the final user of the system.

ALGORITHMS USED



KNN

K-Nearest Neighbours (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN

- a) Lazy learning algorithm – KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
- b) Non-parametric learning algorithm – KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.

Example

The following is an example to understand the concept of K and working of KNN algorithm – Suppose we have a dataset which can be plotted as follows –

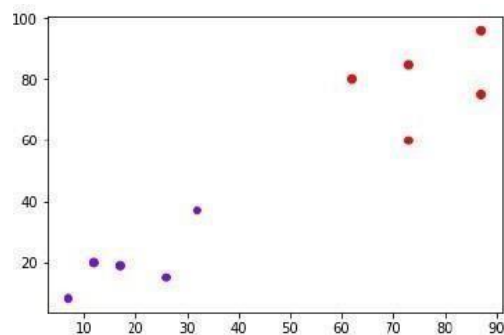


Fig: Plotted Graph of a Dataset

Now, we need to classify new data point with black dot (at point 60, 60) into blue or red class. We are assuming $K = 3$ i.e. it would find three nearest data points. It is shown in the next diagram –

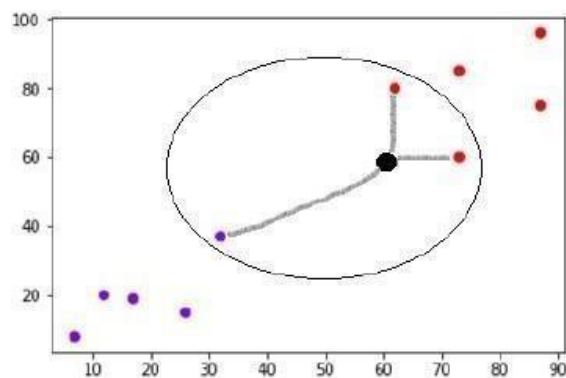


Fig: Classifying new Data point

We can see in the above diagram the three nearest neighbours of the data point with black dot. Among those three, two of them lie in Red class hence the black dot will also be assigned in red class.

1.6 SYSTEM REQUIREMENTS

1.6.1 Software Requirements

One of the most difficult tasks is that, the selection of the software, once system requirement is known that is determining whether a particular software project fits the requirements.

Operating system	Windows8 or Above
Coding Language	Python

1.6.2 Hardware Requirements

The selection of hardware is very important in the existence and proper working of any software.

In the selection of hardware, the size and the capacity requirements are also important.

System	Intel I3 2.4 GHz. Or Advanced
Hard disk	40 GB
Monitor	14' Color Monitor. Or Advanced
RAM	4 GBm
Graphics card	2 GB+

2.REQUIREMENTS

2.1 REQUIREMENTS SPECIFICATION:

2.1.1 Machine learning

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models *of data*. Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data

2.1.2.Of Categories Machine Learning:

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning. Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

2.1.3Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

2.1.4Challenges in Machines Learning

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges.

2.1.5Applications of Machines Learning :

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach.

5.11.1.1 Image classification

5.11.1.2 Video processing

5.11.1.3 Image recognition

2.2 USER INTERFACE:

2.1 Introduction to python:

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system. Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it." Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g. a heterogeneous list be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e. int. long is int as well.
- The division of two integers returns a float instead of an integer. "/" can be used to have the "old" behaviour.
- Text Vs. Data Instead Of Unicode Vs. 8-bit
-

Purpose:

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

3.ANALYSIS:

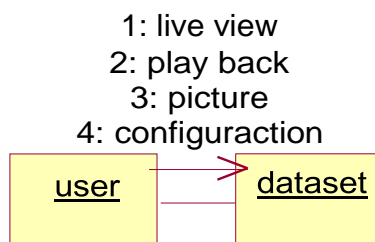
3.1 USE CASE MODEL

3.1.1 User case diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted

3.1.2 Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



4.DESIGN:

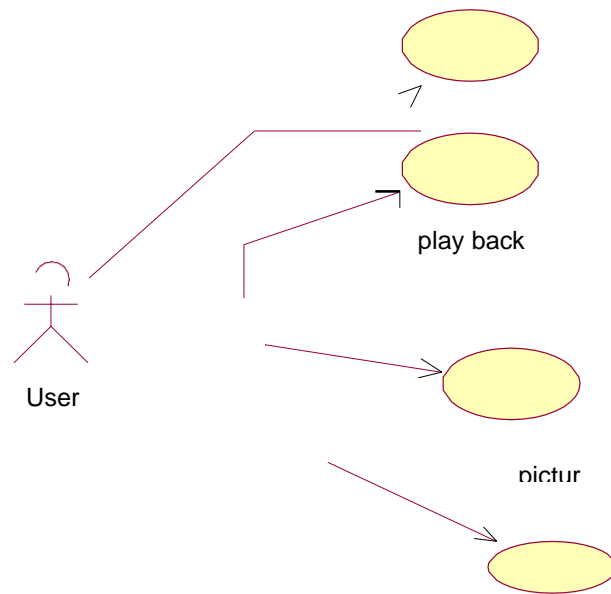
4.1 UML Diagrams for Machine Learning Algorithms

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

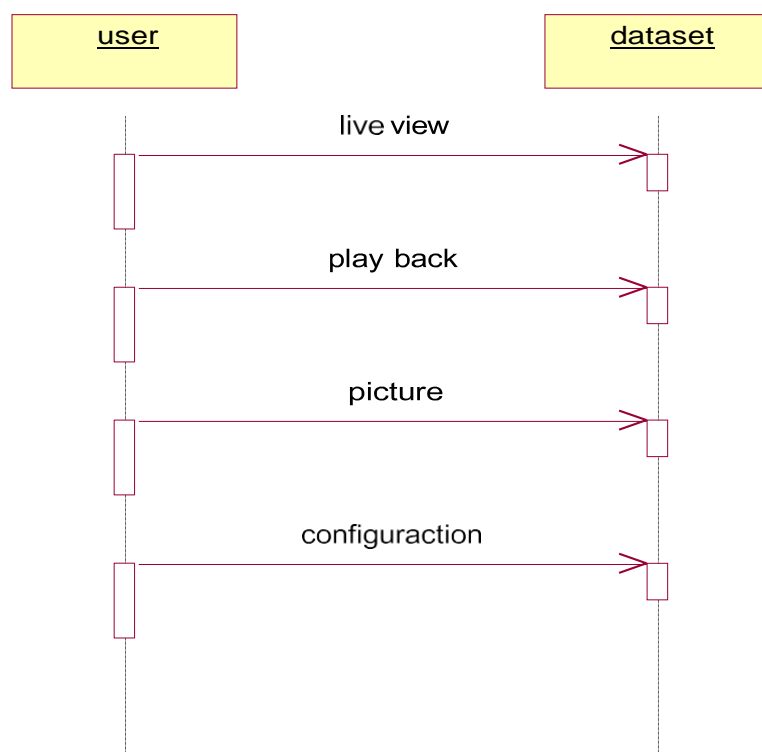
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practice



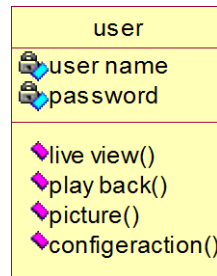
4.2 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



4.3CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type



of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information

4.4COLLABRATION DIAGRAM:

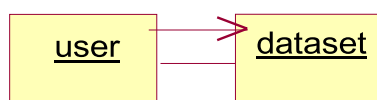
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control

1: live view

2: play back

3: picture

4: configuraction



5.IMPLEMENTATION:

5.1.1Modules Used in Project:

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. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

Numpy:

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas:

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib:

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn:

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

5.11.1 Installation Python:

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

3.1.2 Constraints, prerequisite:

How to Download And Install Python on Windows :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My systemtype is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3.

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click onthe following link: <https://www.python.org>



Now, check for the latest and the correct version for your operating system.

Step 2: Click onthe Download Tab.



Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPC
Gzipped source tarball	Source release		68111671e5b2fb4ae77b9ab01b079be	23617663	305
xz compressed source tarball	Source release		d33e4aa66097051c3eca45ee3604803	37131432	305
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7563da71a442c8a8ce08e6	34898436	305
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5d505c38217a45773b9eae936d343f	38082845	305
Windows .hug file	Windows		d63999573a2c9832ac58cad6b4ef7cd2	8131761	305
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/v64	9609c3c76d8ee3b6a6e3184ae072ba2	7504391	305
Windows x86-64 executable installer	Windows	for AMD64/EM64T/v64	a702bcb0ad70d65dc35c3a83e5d3400	26480368	305
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/v64	28c31c088b6f7aee651a3ba351b4bd2	1362904	305
Windows x86 embeddable zip file	Windows		9fab38d198a1d79fda941357413bd8	6741626	305
Windows x86 executable installer	Windows		33c3802942a5446a386451476394788	25663848	305
Windows x86 web-based installer	Windows		1b670cfad3117d82c309f3ea371d87c	1324608	305

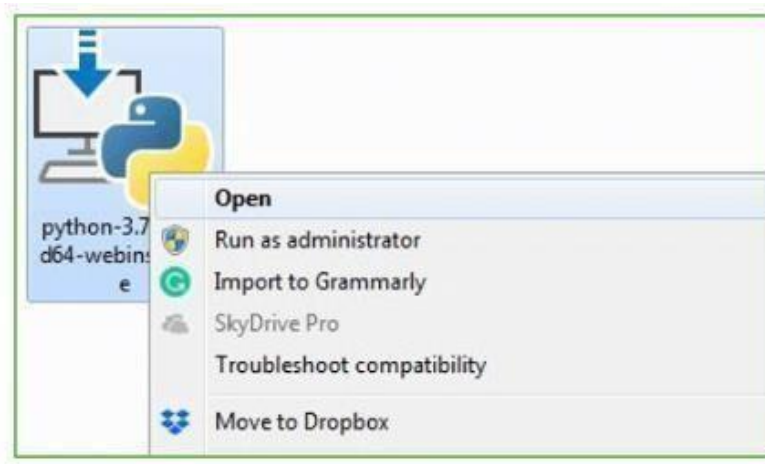
- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



Step 3: Click on Install NOW After the installation is successful. Click on Close.



With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.



Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python -V** and press Enter.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\DELL>python -V
Python 3.7.4
C:\Users\DELL>_
```

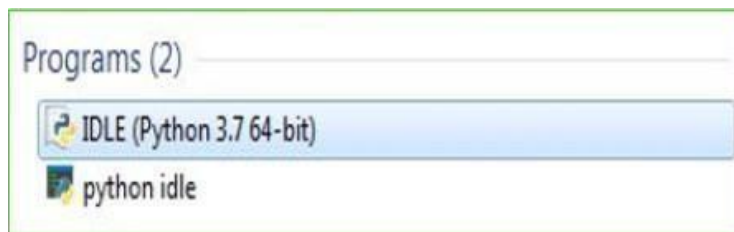
Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

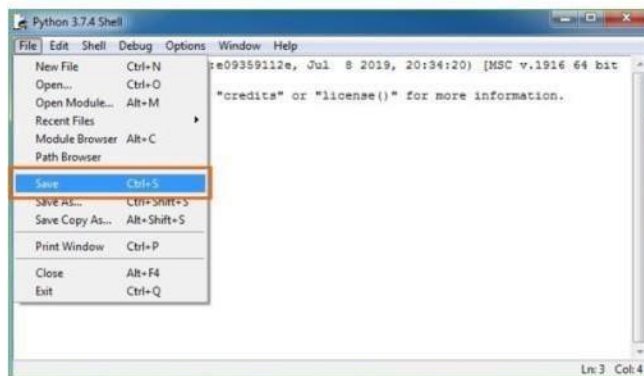
Step 1: Click on Start

Step 2: In the Windows Run command, type “python idle”.



Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



Step 5: Name the file and save as type should be Python files. Click on SAVE.

Here I have named the files as Hey World.**Step 6:** Now for e.g. **enter print**

6 .USER SCREEN

6.1Suspicious Activity Detection

In this project we need to detect person behaviour as suspicious or not, now a day's every where CCTV cameras are installed which capture videos and store at centralized server and manually scanning those videos to detect suspicious activity from human required lots of human efforts and time. To overcome from such issue author is asking to automate such process using Machine Learning Algorithms.

To automate that process first we need to build training model using huge amount of images (all possible images which describe features of suspicious activities) and 'Convolution Neural Network' using TENSOR FLOW Python module.

Then we can upload any video and then application will extract frames from uploaded video and then that frame will be applied on train model to predict its class such as 'suspicious or normal'.

To implement above concept we need to install python 3.5 version in 64 bit laptop. I will send this software with code. While software installation u need to select checkbox saying add path to system variable. This option will show on first or second screen of installation. Once you install software execute below commands. Your system must connect to internet.

```
pip install tensorflow pip install numpy pip install scipy
```

```
pip install opencv-python pip install pillow
```

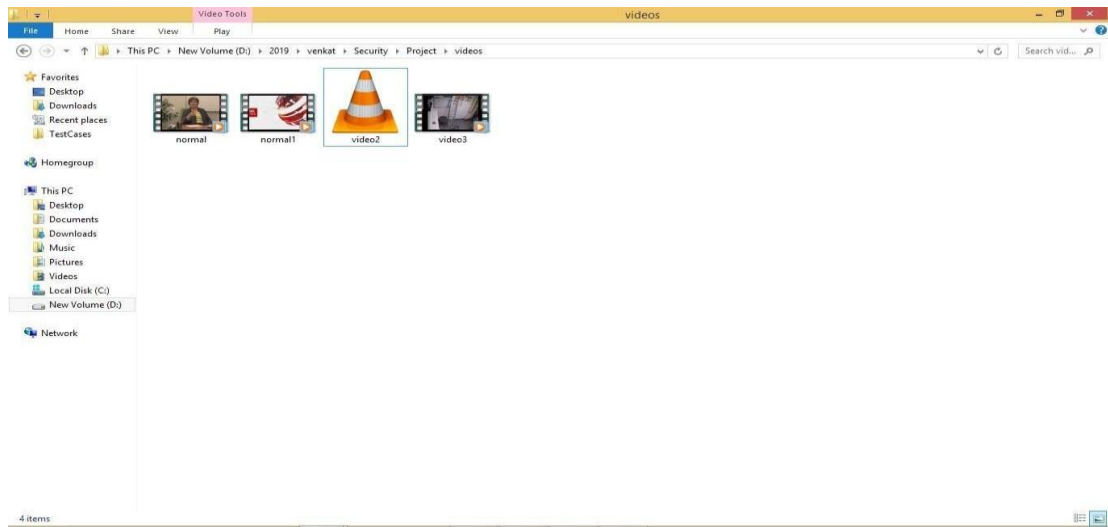
```
pip install matplotlib pip install h5py
```

```
pip install keras
```

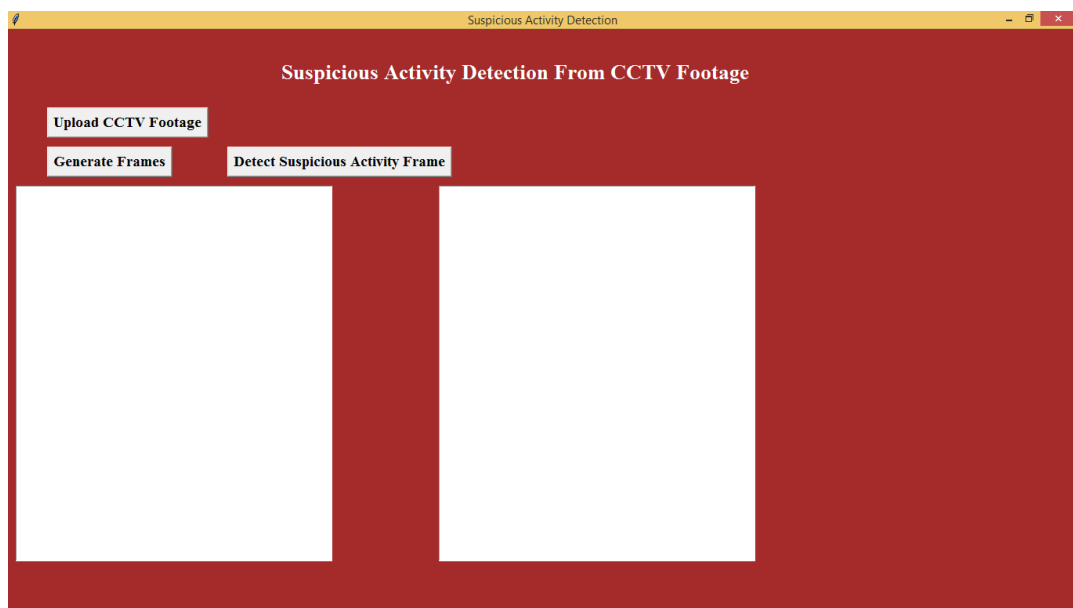
```
pip install
```

<https://github.com/OlafenwaMoses/ImageAI/releases/download/2.0.2/imageai-2.0.2-py3-none-any.whl>

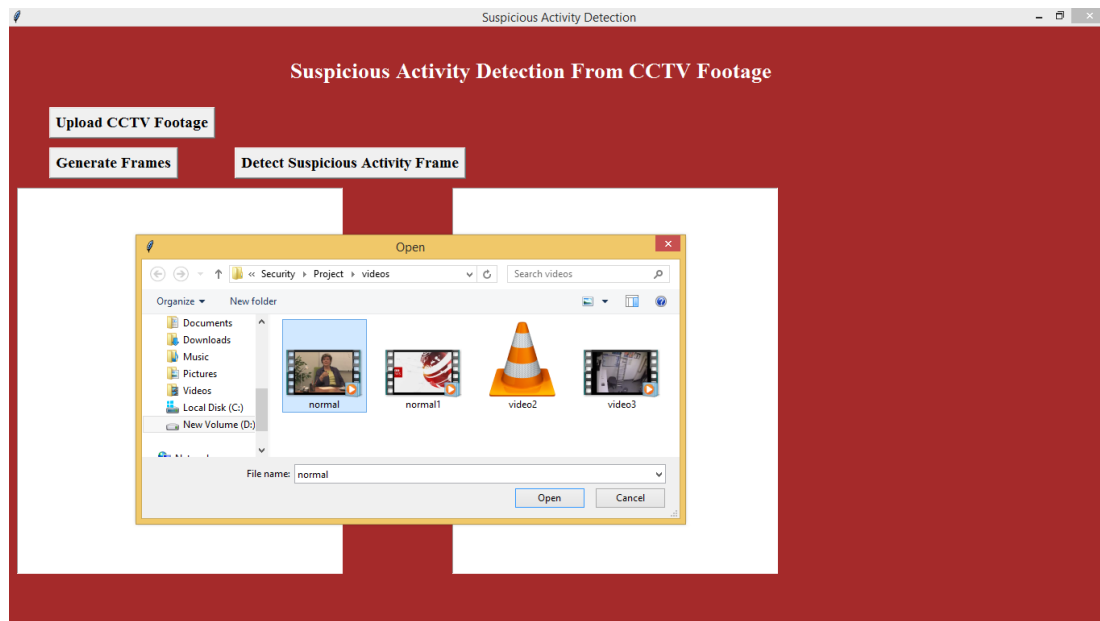
After executing above commands all software will be installed. To monitor i am using below videos. For training i used human images who cover their faces to perform suspicious activity and if any video contains person covering their faces then application will detect it as a suspicious activity



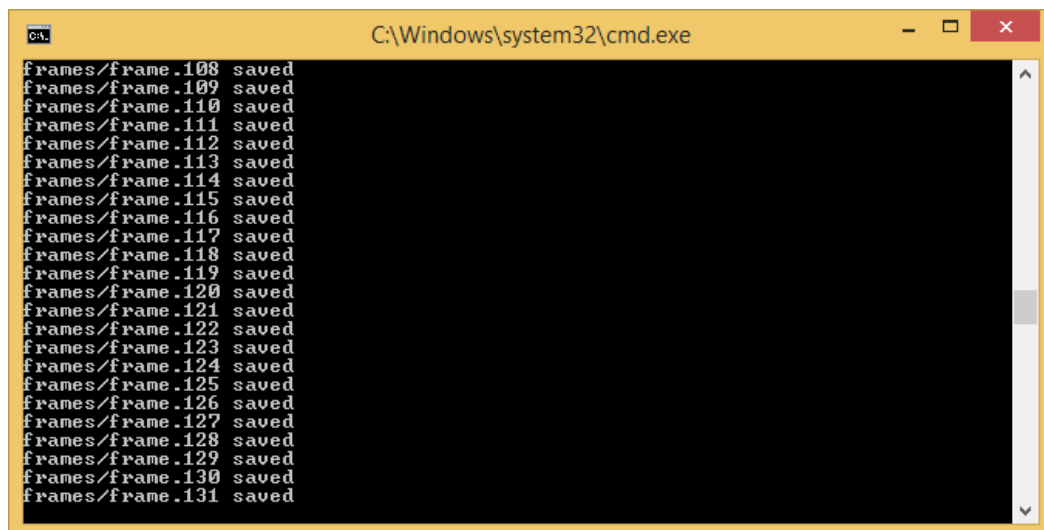
Double click on 'run.bat' file from project folder to start project execution. We will get below screen



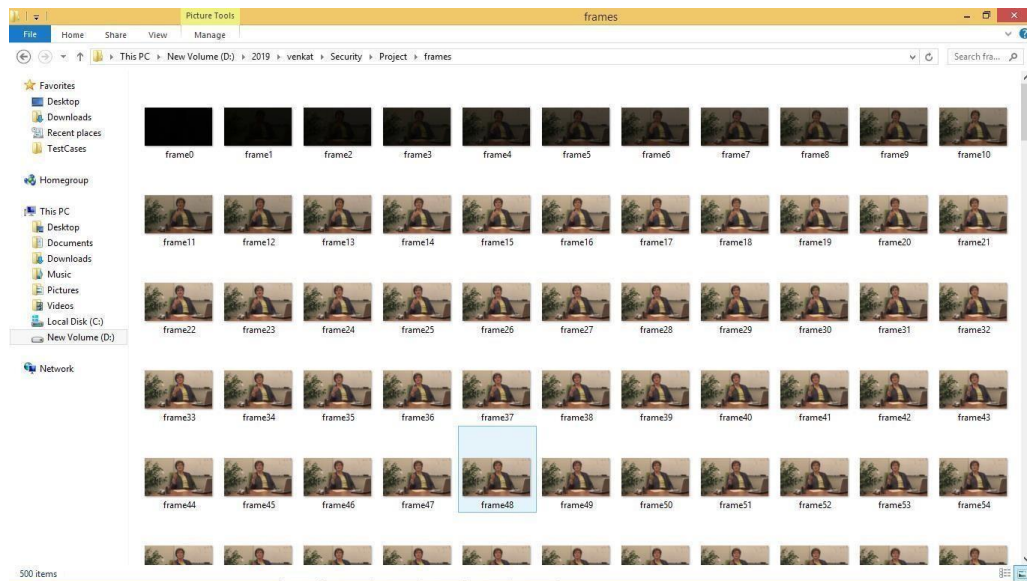
Click on 'Upload CCTV Footage' button to upload video



In above screen i am uploading one normal video. After uploading video click on 'Generate Frames' button to generate frame



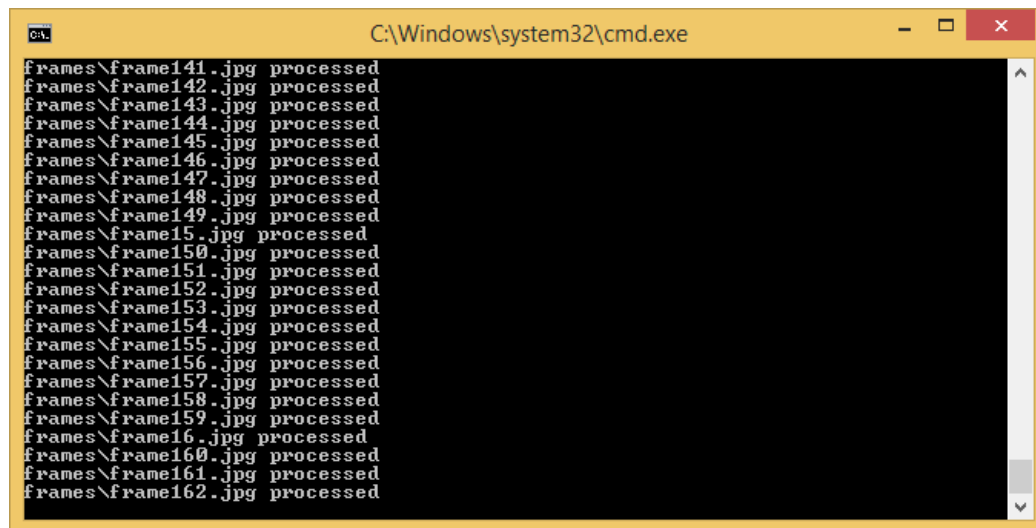
In above black screen we can see extracted frames are saving inside 'frames' folder frame no. Now we see frames folder below which has images from video.



In above folder screen we can see all images from video extracted. After frame extraction will get below screen.



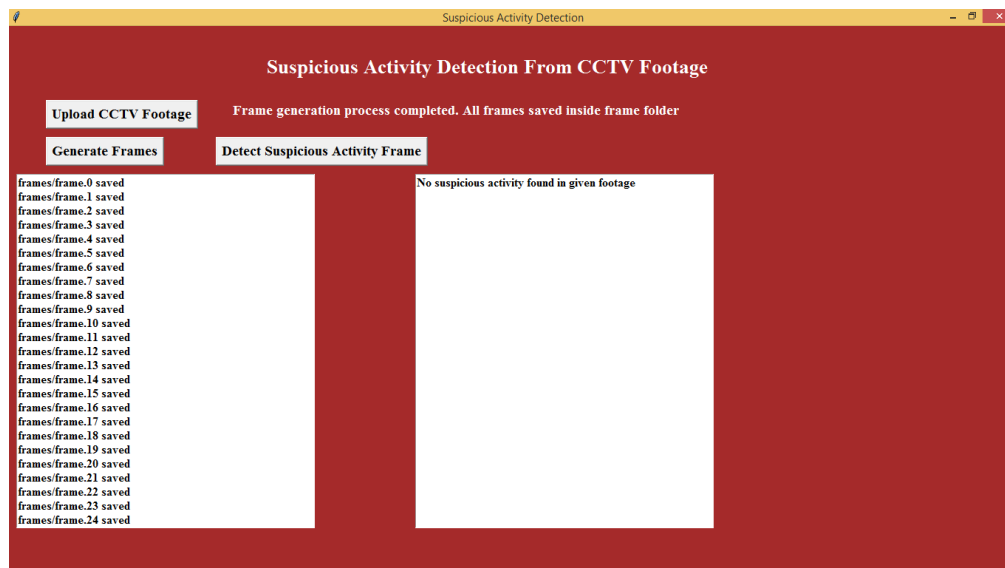
Now click on 'Detect Suspicious Activity Frame' button to start monitoring frames for suspicious activity.



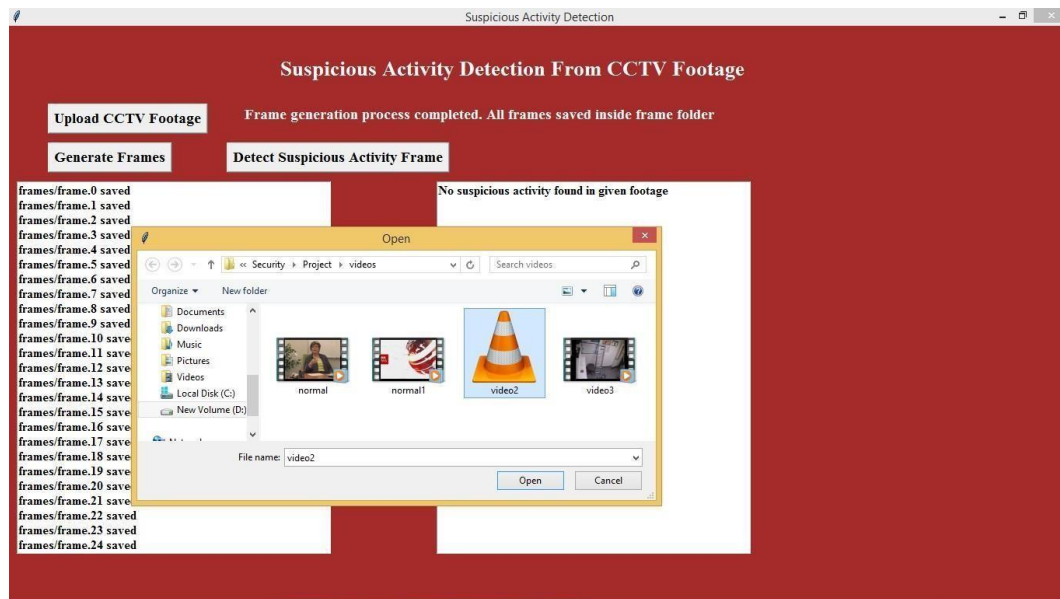
```
C:\Windows\system32\cmd.exe

frames\frame141.jpg processed
frames\frame142.jpg processed
frames\frame143.jpg processed
frames\frame144.jpg processed
frames\frame145.jpg processed
frames\frame146.jpg processed
frames\frame147.jpg processed
frames\frame148.jpg processed
frames\frame149.jpg processed
frames\frame150.jpg processed
frames\frame151.jpg processed
frames\frame152.jpg processed
frames\frame153.jpg processed
frames\frame154.jpg processed
frames\frame155.jpg processed
frames\frame156.jpg processed
frames\frame157.jpg processed
frames\frame158.jpg processed
frames\frame159.jpg processed
frames\frame160.jpg processed
frames\frame161.jpg processed
frames\frame162.jpg processed
```

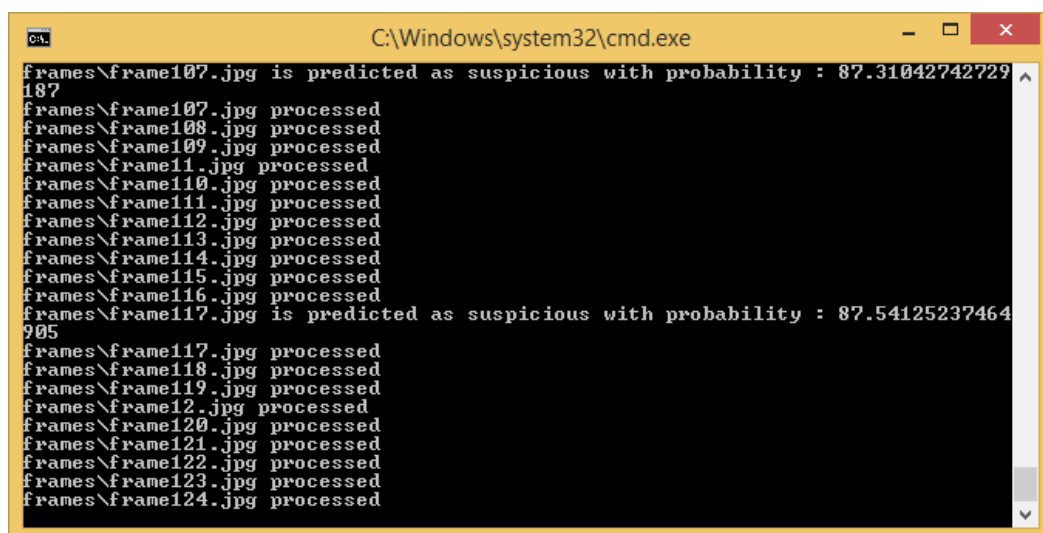
In above black console window we can see processing of each frame to detect suspicious activity.



In above screen we can see frames scanned and no suspicious activity found. Now we will upload another video and check status.

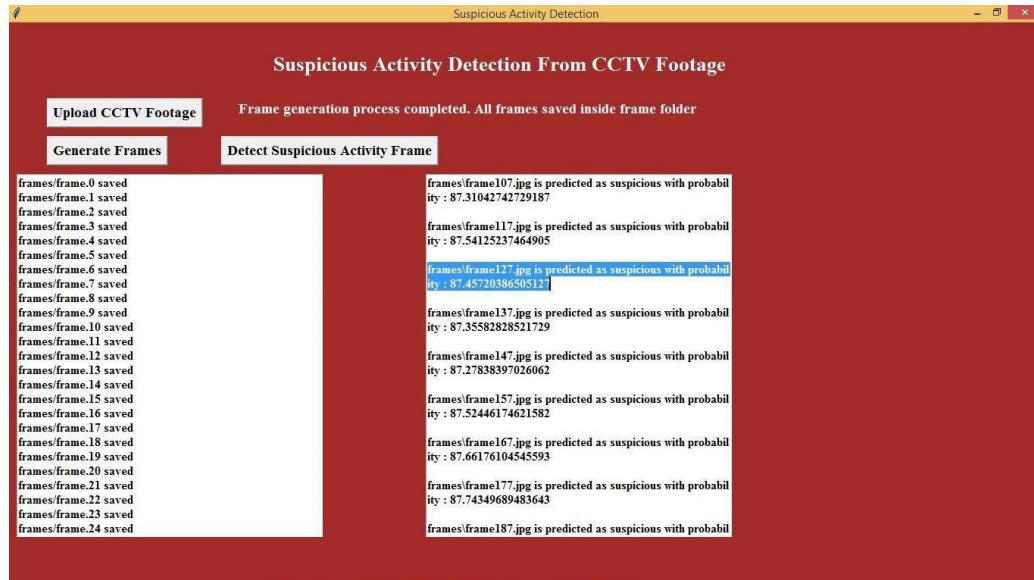


In above screen we should upload 'Video2' and then extract frames



In above screen for uploaded video we can see suspicious activity found at frame117.jpg. After scanning all images we will get below details screen. Now in below screen we can see frame117.jpg image from frames folder.

In above screen frame117 showing one image of a person with face covering. Similarly we can see all frames details in below screen which has such activities



In above screen in right text area we can see details of all frames



Note: you to can upload your own videos and check but your videos must have person covering their faces or doing shoplifting robbers videos. Your videos must be like similar one which i used in this project

CODING

```
from tkinter import messagebox

from tkinter import *
from tkinter import simpledialog

import tkinter
from tkinter import filedialog from imutils
import paths import matplotlib.pyplot as plt
import datetime
from tkinter.filedialog import askopenfilename
import cv2 import shutil import os
from imageai.Prediction.Custom
import CustomImagePrediction
import os
main = tkinter.Tk()
main.title("Suspicious Activity Detection") main.geometry("1200x1200")

global filename

execution_path = os.getcwd() prediction = CustomImagePrediction()
prediction.setModelTypeAsResNet() prediction.setModelPath("model.h5")
prediction.setJsonPath("model_class.json") prediction.loadModel(num_objects=2)

def upload(): global filename
filename = askopenfilename(initialdir = "videos")
pathlabel.config(text=filename)

def generateFrame():
global filename text.delete('1.0', END)
if not os.path.exists('frames'): os.mkdir('frames')
else:
shutil.rmtree('frames') os.mkdir('frames')
vidObj = cv2.VideoCapture(filename) count = 0
success = 1 while success:
```

```

success, image = vidObj.read() if count < 500:
cv2.imwrite("frames/frame%d.jpg"
% count, image)

```

```

text.insert(END,"frames/frame."+str(count
)+" saved\n")
print("frames/frame."+str(count)+" saved")

```

```

#pathlabel.config(text="frames/frame."+str
(count)+" saved")
else:
break count += 1
pathlabel.config(text="Frame    generation
process completed. All frames saved inside
frame folder")

```

```

def detectActivity(): imagePaths
=
sorted(list(paths.list_images("frames")))
count = 0
option = 0;
text1.delete('1.0', END)
for imagePath in imagePaths:
predictions, probabilities =
prediction.predictImage(imagePath,
result_count=1)
for eachPrediction, eachProbability in
zip(predictions, probabilities):
if float(eachProbability) > 80: count =
count + 1;
if float(eachProbability) < 80: count = 0
if count > 10:
option = 1
print(imagePath+" is predicted as
"+eachPrediction+" with probability : "
+str(eachProbability))

```

```
text1.insert(END,imagePath+" is predicted  
as "+eachPrediction+" with probability : "  
+str(eachProbability)+"\n\n")  
count = 0;  
print(imagePath+" processed") if  
option == 0:  
text1.insert(END,"No suspicious  
activity found in given footage")
```

```
font = ('times', 20, 'bold')  
title = Label(main, text='Suspicious  
Activity Detection From CCTV Footage')  
title.config(bg='brown', fg='white')  
title.config(font=font)  
title.config(height=3, width=80)  
title.place(x=5,y=5)
```

```
font1 = ('times', 14, 'bold')  
upload = Button(main, text="Upload  
CCTV Footage", command=upload)  
upload.place(x=50,y=100)  
upload.config(font=font1)
```

```
pathlabel = Label(main)  
pathlabel.config(bg='brown', fg='white')  
pathlabel.config(font=font1)  
pathlabel.place(x=300,y=100)
```

```
depthbutton = Button(main,  
text="Generate Frames",  
command=generateFrame)  
depthbutton.place(x=50,y=150)  
depthbutton.config(font=font1)
```

```
userinterest = Button(main, text="Detect
```

```
Suspicious Activity Frame",  
command=detectActivity)  
userinterest.place(x=280,y=150)  
userinterest.config(font=font1)
```

```
font1 = ('times', 12, 'bold')  
text=Text(main,height=25,width=50)  
scroll=Scrollbar(text)  
text.configure(yscrollcommand=scroll.set)  
text.place(x=10,y=200)  
text.config(font=font1)
```

```
text1=Text(main,height=25,width=50)  
scroll=Scrollbar(text1)  
text1.configure(yscrollcommand=scroll.set  
)  
text1.place(x=550,y=200)  
text1.config(font=font1)  
main.config(bg='brown')  
main.mainloop()
```

7.TESTING METHODOLOGY

7.1SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.2 TYPES OF TESTING

7.2.1 Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2.2 Integration testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.3FUNCTIONAL TESTING:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals

7.3.1 Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output: identified classes of application outputs must be exercised. Systems/Procedures :
interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.4 White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level

7.5 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered

8.CONCLUSION

The results of these tests show, that it is not possible to rule out the possibility, that changes in parameters, such as framerate or resolution, will influence the detection capabilities of any surveillance system. It can be observed that systems work differently. Most of the systems compared showed signs that they use data for motion detection which are not influenced by current settings of the system. It is very likely that these systems have a dedicated video stream used for analysis (such as video- based motion detection). Differences in motion detection success rates between individual devices are caused by non-uniform sensitivity settings for motion detection. However, these changes have no influence on the tested data, because the changes would show in both lower and higher success rates.

9.REFERENCES

1. *EDIS Publishing*: [www.edis.sk](<http://www.edis.sk>) - Search for the books by LOVEČEK et al.
2. *Elsevier*: [www.elsevier.com](<http://www.elsevier.com>) - Look for "Digital Video Surveillance and Security" by Caputo.
3. *Taylor & Francis*: [www.routledge.com](<http://www.routledge.com>) - Search for "Intelligent Network Video" by Nilson.
4. *ISO Standards*: [www.iso.org](<http://www.iso.org>) - Search for STN EN standards related to video surveillance systems.
5. *ResearchGate*: [www.researchgate.net](<http://www.researchgate.net>) - You may find articles or research related to these topics.