

# **SUSPICIOUS ACTIVITY DETECTION**

## **Abstract**

In current day's suspicious event detection has become one of the critical problem for the cyber department or police department. In general this is mainly found by using either manual observance of CCTV camera recording and then try to predict whether any suspicious activity present in that live recorded video. In general manual method of detecting the suspicious activity may not give accurate results. 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'.

# **CHAPTER 01**

## **INTRODUCTION**

# 1. INTRODUCTION

Human face and human behavioural pattern play an important role in person identification. Visual information is a key source for such identifications. Surveillance videos provide such visual information which can be viewed as live videos, or it can be played back for future references. The recent trend of 'automation' has its impact even in the field of video analytics. Video analytics can be used for a wide variety of applications like motion detection, human activity prediction, person identification, abnormal activity recognition, vehicle counting, people counting at crowded places, etc. In this domain, the two factors which are used for person identification are technically termed as face recognition and gait recognition respectively. Among these two techniques, face recognition is more versatile for automated person identification through surveillance videos. Face recognition can be used to predict the orientation of a person's head, which in turn will help to predict a person's behaviour. Motion recognition with face recognition is very useful in many applications such as verification of a person, identification of a person and detecting presence or absence of a person at a specific place and time. In addition, human interactions such as subtle contact among two individuals, head motion detection, hand gesture recognition and estimation are used to devise a system that can identify and recognize suspicious behaviour among pupil in an examination hall successfully. This paper provides a methodology for suspicious human activity detection through face recognition.

Video processing is used in two main domains such as security and research. Such a technology uses intelligent algorithms to monitor live videos. Computational complexities and time complexities are some of the key factors while designing a real-time system. The system which uses an algorithm with a relatively lower time complexity, using less hardware resources and which produces good results will be more useful for time-critical applications like bank robbery detection, patient monitoring system, detecting and reporting suspicious activities at the railway station, etc

## 1.1 Aim Of The Project

Manual monitoring of exam hall through invigilators and manual monitoring of exam hall through surveillance videos is performed throughout the world. Monitoring an examination hall is a very challenging task in terms of man power. Manual monitoring of examination halls may be prone to error during human supervision. Such a system when implemented as

an ‘automatic suspicious activity detection system’ will not only help in detecting suspicious activities but also helps in minimizing such activities. Moreover, the probability of error will be much lesser. This system will serve as a useful surveillance system for educational institutions.

## **1.2 Methodology**

The Suspicious Human Activity Detection system aims to identify the students who indulge in malpractices/suspicious activities during the course of an examination. The system automatically detects suspicious activities and alerts administration.

## **1.3 SignificanceOf The Work**

This proposed application describes a technology in which real time videos are analysed and are used for human activity analysis in an examination hall, thus helping to classify whether the particular person’s activity is suspicious or not. The system developed identifies abnormal head motions, thereby prohibiting copying. It also identifies a student moving out of his place or swapping his position with another student. Finally the system detects contact between students and hence prevents passing incriminating material among students. In our research, we have contributed upon a system that will intellectually process live video of examination halls with students and classify their activities as suspicious or not. This research proposes an intelligent algorithm that can monitor and analyse the activities of students in an examination hall and can alert the educational institute’s administration on account of any malpractices/suspicious activities.

## **1.4Report Organization**

In this report we try to organize the project into chapter wise and arranged as per software development life cycle. The chapter 1 contains the introduction about the project and in chapter 2 we try to discuss about the literature survey. In chapter 3 we try to arrange the design aspects which are discussed about this project. In chapter 4,we discuss about implementation aspects present in our application. In chapter 5,we discuss about experimental results and finally end with reports and conclusion and references as last topics.

# **CHAPTER 02**

## **LITERATURE SURVEY**

# LITERATURE SURVEY

## 2.1 Introduction

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, the next steps are to determine which operating system and language used for developing the tool. Once the programmers start building the tool, the programmers need lot of external support. This support obtained from senior programmers, from book or from websites. Before building the system the above consideration is taken into for developing the proposed system.

## 2.2 Related Work

R. K. Tripathi et al. fire detection. An overview of the progress in the past decade in the field of abandoned object detection, theft detection, falling detection, violence activity detection and fire detection discussed in Sect. 4. The general framework for suspicious activities detection is discussed in Sect. 5. Section 6 presents the Datasets and Evaluation measures used for abandoned object detection, theft detection, falling detection, accidents and illegal parking detection on road, violence activity detection and fire detection. Finally, the last section presents conclusion and future work.

## Motivation and applications

Importance of the suspicious human activities recognition from video surveillance is to prevent the theft cases, leaving abandoned objects for the explosive attacks by terrorists, vandalism, fighting and personal attacks and fire in the different highly sensitive areas such as banks, hospitals, malls, parking lots, bus and railway stations, airports, refineries, nuclear power plants, schools, university campuses, borders etc. Intelligent video surveillance protects the following areas from suspicious activities (Yilmaz et al. 2006): University campus and academic institutions Video surveillance is being used in university campuses and other academic

institutions to monitor the activities of students for the safety of assets from theft and vandalism. It also helps to prevent the inappropriate behavior of the students and fighting among the students. It also monitors the perimeter of the university campus, school and academic institutions for the safety of the students and faculties. Video surveillance can be used at the time of examination to monitor the suspicious activity of the students in the examination hall.

### **Public infrastructure**

To save population and public infrastructures such as borders, laboratories, prisons, military bases, temples, parking lots; video surveillance is helpful to prevent the theft, vandalism, fighting and personal attacks, increasing crowd, explosive attacks.

### **Retail trade**

This is a growing market for the use of video surveillance to detect the suspicious human activity for both the internal such as warehouses, stores and external like parking lots security. Even the small shops are utilizing the cameras to monitor the human activities and to capture the video evidence in case of theft or an incident. In chain stores, much more sophisticated video surveillance systems are set up for centralized monitoring of different locations. Suspicious activity recognition from video surveillance helps to monitor employee fraud and theft, monitor wares and inventory, protecting material goods and infrastructures, protecting staff and clients, monitoring parking lots, vehicles, entries and exits, and emergency situations such as fire.

### **Airports**

Airports are high security sensitive areas where the safety of passengers, runway and airplane is the most important in any country. Real-time suspicious human activity recognition system from video surveillance provides high level security to such security sensitive areas.

### **Railway and bus stations**



The use of video surveillance at railway and bus stations plays a vital role in case of monitoring platforms, routes, parking lots, rails and tunnels. These areas are the prime targets of the terrorists for explosive attacks by leaving a bag containing bomb. Suspicious activity recognition system from video surveillance can recognize the abandoned object and can alarm to remove it from public place for the protection of passengers, personnel and infrastructures.

### **Banking sector**

Video surveillance plays an important role in banking sectors to provide the security. The presence of cameras prevents committing the armed robbery and assault. Automated bank machines are prime targets for criminal acts. Surveillance camera helps to detect fraud, for example; the installation of a device to read the magnetic information on bank cards. Intelligent video surveillance can increase monitoring effectiveness in banking sectors. It provides monitoring to all the branches in order to detect suspicious behavior. In ATMs, it also helps to prevent theft cases. Gaming industries and casinos Suspicious activity recognition from video surveillance can help to detect the cheating, heists, and other crimes. Since monitoring of casino requires watching the activity of human beings in a crowded environment, intelligent video surveillance is an interesting way of helping security personnel.

### **Hospitals**

Video surveillance can also be used in hospitals to monitor the patients at home to monitor elder people or children. It can even be found in ambulances to monitor a patient remotely. Video surveillance can monitor the activity of the patients in hospitals and can recognize the suspicious activity such as vomiting, fainting and other unusual activity of the patients.

# **CHAPTER 03**

## **DESIGN ASPECTS**

### **SYSTEM ANALYSIS**

### **3.1 Feasibility Study**

The feasibility of the project 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

#### **3.1.1 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.

#### **3.1.2 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.

#### **3.1.3 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 to educate the user about the system and to make him familiar with it.

His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

### **3.2 Existing System**

In the existing system all the primitive methods try to identify the suspicious user activity based on 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. This may give some times correct assumption and some times this may generate in-accurate results.

### **Limitation Of Existing System**

The following are the main limitations of the existing system. They are as follows:

- 1) In the existing clouds there is no method like Automatic detection of suspicious activity of human from CCTV recording video.
- 2) All the data is accessed by centralized server and hence if the hacker or intruder hack the centralized location, then the data may get modified as per the attacker wish.
- 3) All the existing systems failed to utilize the property of ML and CNN models to detect the suspicious activity automatically and then find out the culprit.
- 4) There is no accurate and efficient results in primitive systems.

### **3.3 Proposed System**

In this proposed system we automate that process of finding the suspicious activity based on real time video captured and recorded from the CCTV web camera. We need to build training model using huge amount of images (all possible images which describe features of

suspicious activities) and then try to construct a CNN Model (Convolution Neural Network) using TENSOR FLOW Python module. Once the CNN model is deployed 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’. Here we can get accurate results from the recorded video without any disturbance.

### **3.3.1 Advantages of the Proposed System**

The following are the advantages of the proposed system :

1. The comparison results demonstrate that our scheme has advantages in terms of efficiency and functionality, and thus is more feasible for practical applications.
2. The proposed model is very accurate in classifying the video sequence containing the suspicious activity and normal activity very easily.
3. The proposed CNN model can achieve high level of accuracy when compared with manual approach of detecting the suspicious activity.
4. The proposed CNN model using tensor flow is very efficient for detecting the suspicious activity.

### **3.4 System Design**

Software System design is an interactive process through which the requirements are translated into “blue print” for constructing the software. The three characteristics that are the evaluation of a good design of the software are as follows:

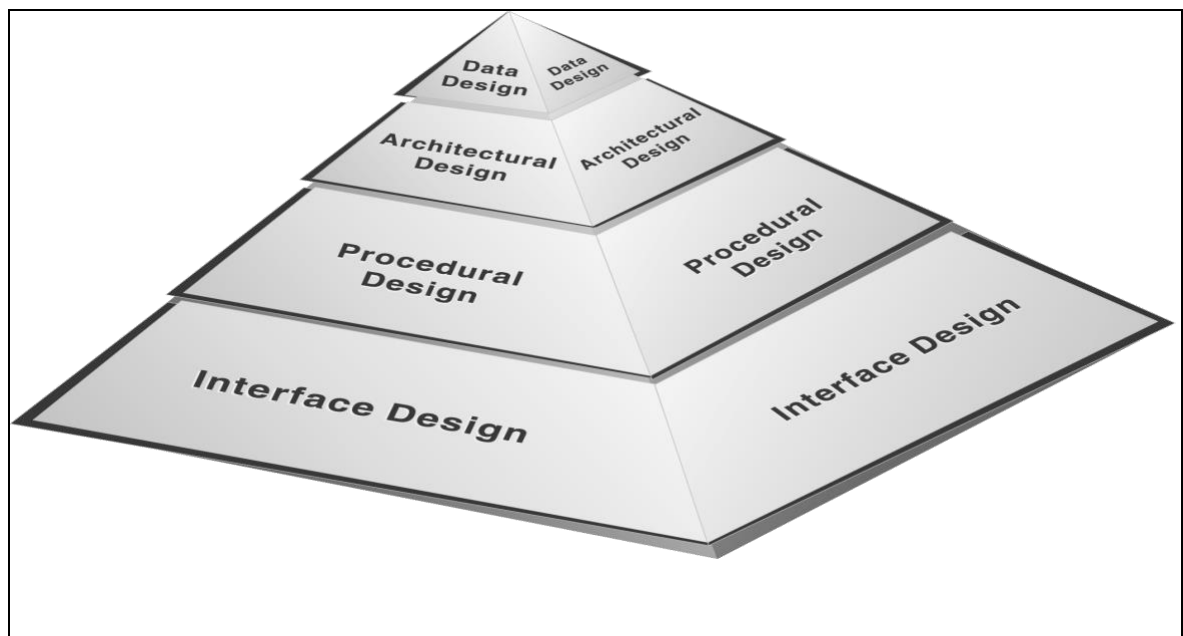
1. The design must implement the entire explicit requirement contained in the analysis mode and it must be accommodate the entire implicit requirement desired by the user.
2. The design must reliable and act as an understandable guide for those, who generate code and text subsequently maintain the software.

3. The design should provide a complete picture of the software, addressing the data functional and behavioral domain from an implementing perspective

The design-phase helps in providing the procedural details necessary for implementing the system recommended in the feasibility study. The emphasis is on translating the performance requirements into design specification. Design phase goes through logical and physical phases of the development.

System Design is a process in which the end users requirements are transformed into software/package and a specification for a computer based information-processing system. System is a solution, on “how to” approach to the creation of a new system. It describes the solution to the problem and serves as the starting point for programming.

Software design is both a process and a model the design process is a set of interactive Steps that enable the designer to describe all aspects of the software to be built. Figure 3.1 as shown above describes the interactive steps for system design for constructing the software. Basic design principles enable the software’s engineer to navigate the design process.



**Fig.3.1 Steps for System Design**

Software design is both a process and a model the design process is a set of interactive Steps that enable the designer to describe all aspects of the software to be built. Figure 3.1 as shown above describes the interactive steps for system design for constructing the software. Basic design principles enable the software's engineer to navigate the design process.

Some of the design principles are:

- The design should be traceable.
- The design should not reinvent the wheel.
- The design should “minimize the intellectual distance “between the software’s and the problem as it exist in the real world”.
- The design should exhibit uniformly and integration.

### **3.5 Data Flow Diagram**

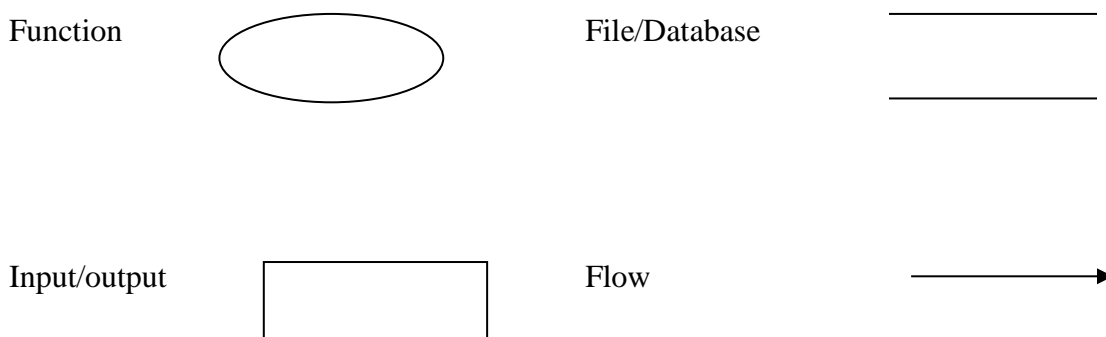
A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change or transform data throughout a system. It's a structured analysis and design tool that can be used for flowcharting in place of, or in association with, information oriented and process oriented system flowcharts. When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply a physical implementation. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams.

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

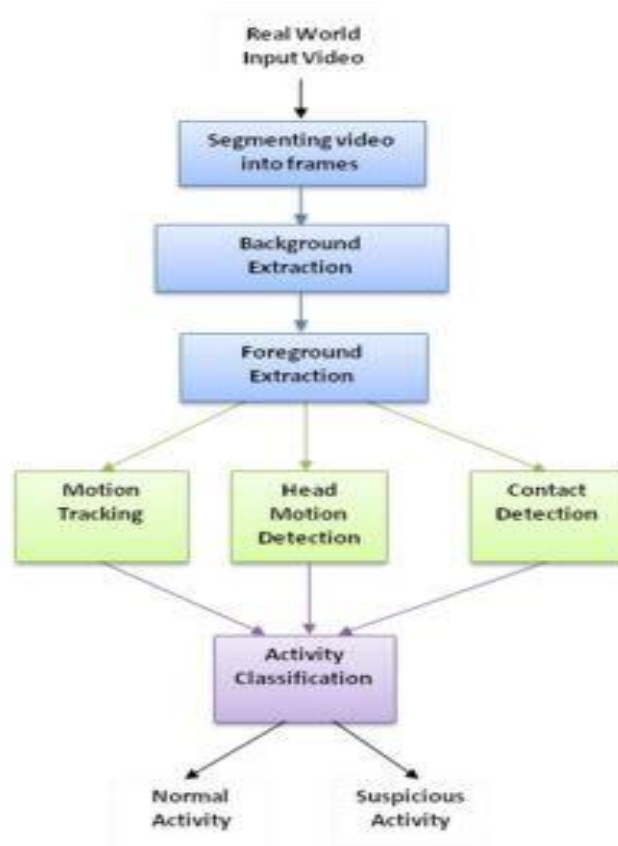
Figure 3.2 show four basic symbols that are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes. The figure 3.3 shows the data flow diagram of our project.

### ➤ SYMBOLS



*Fig. 3.2 DFD Elements*





**Fig.3.3 Dataflow Diagram for Proposed Application**

### 3.6UML Diagrams

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.

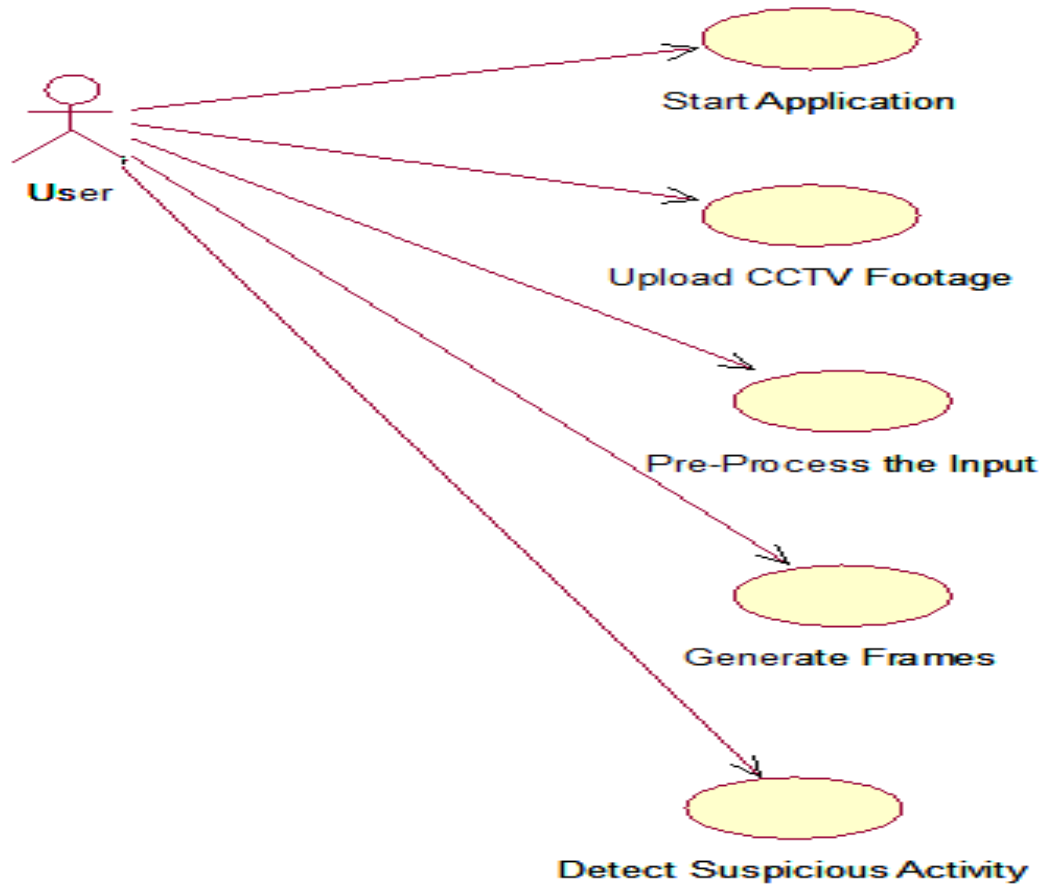
The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the 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 practices.

### **3.6.1 Use Case Diagram**

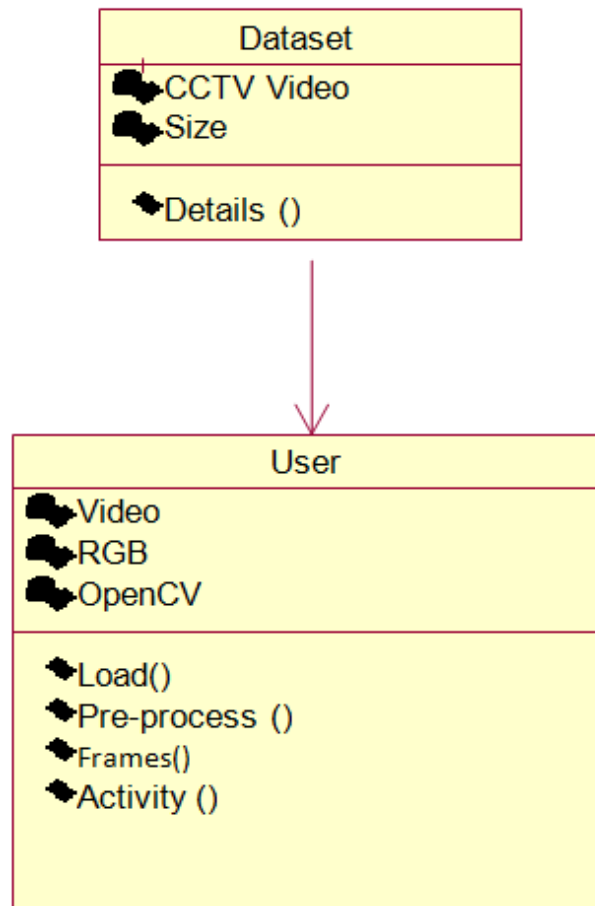
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies 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.



**Fig.3.6.1 Use Case Diagram**

### **3.6.2 Class Diagram**

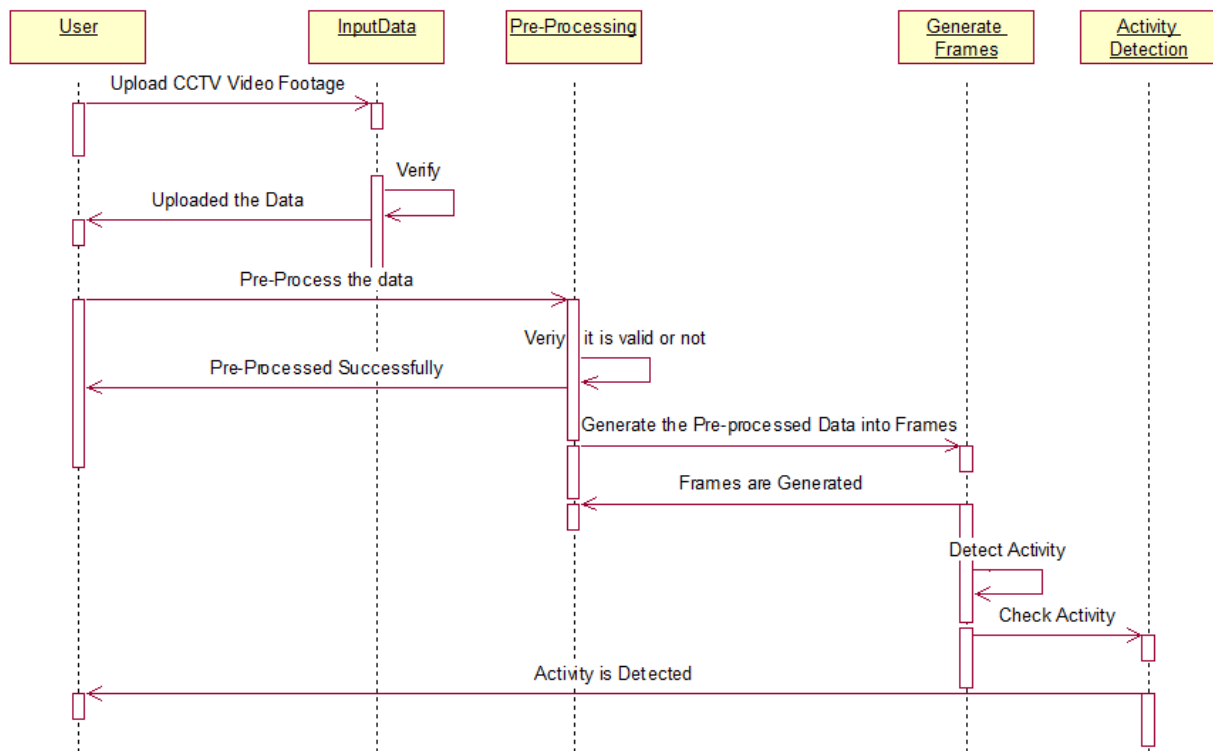
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**Fig.3.6.2 Class Diagram**

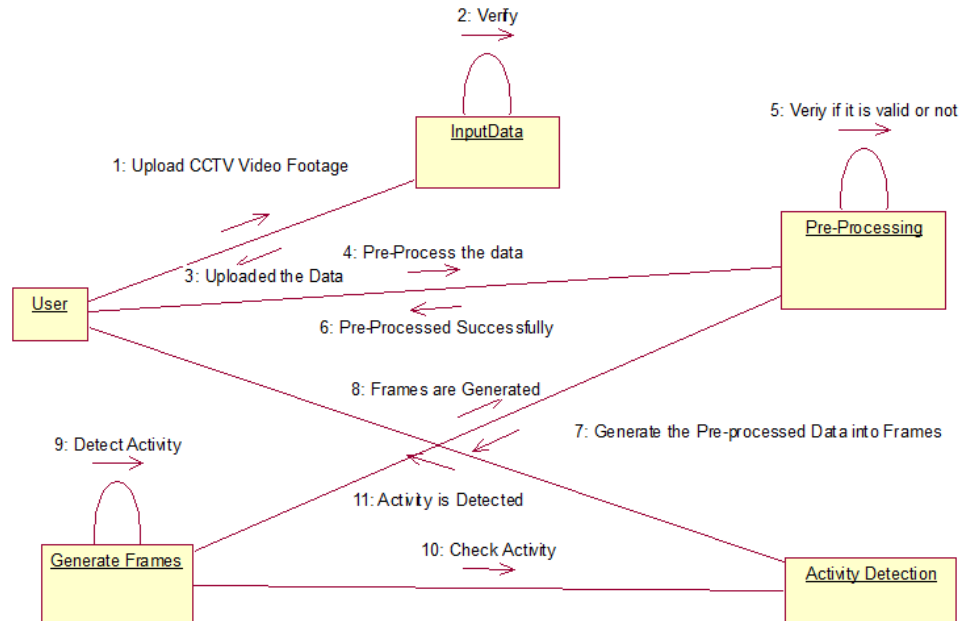
### 3.6.3 Sequence Diagram

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.



**Fig.3.6.3 Sequence Diagram**

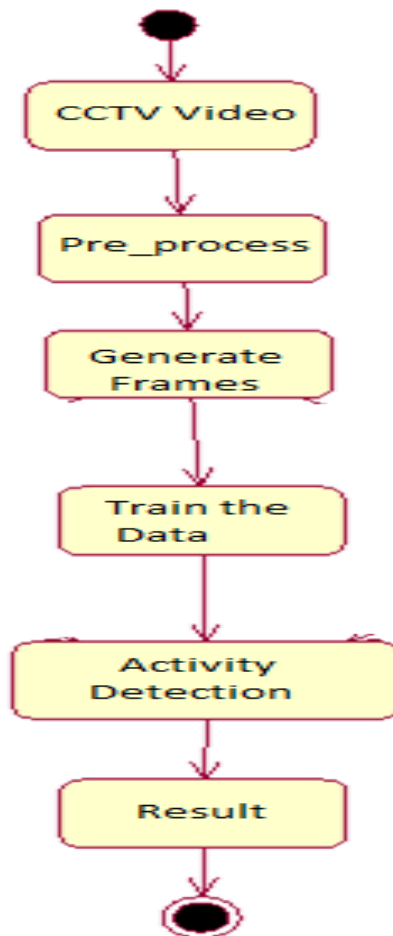
### 3.6.4 Collaboration Diagram



**Fig.3.6.4 Collaboration Diagram**

### 3.6.5 Activity Diagram

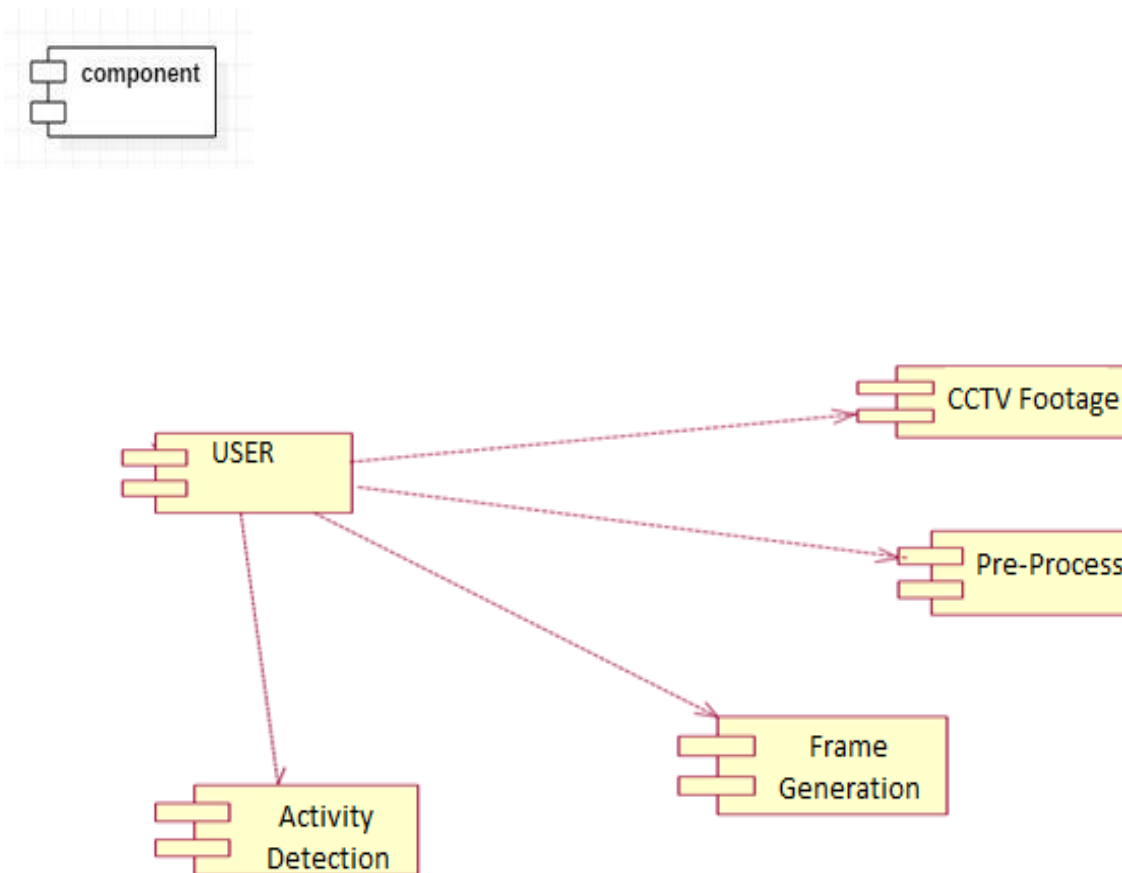
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.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



**Fig.3.6.5 Activity diagram**

### 3.6.6 Component Diagram

Component diagrams represent a set of components and their relationship. These components consist of classes, interfaces, or collaborations. It is a special kind of diagram in UML. The purpose is also different from all other diagrams, It does not describe the functionality of the system but it describes the components used to make those functionalities. Component diagrams represent the implementation view of system.



**Figure 3.6.6 Component Diagram**



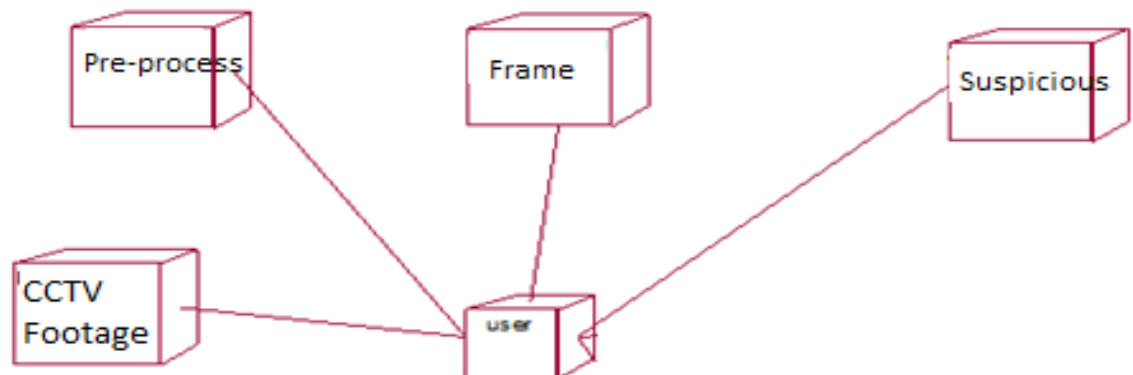
### 3.6.7 Deployment Diagram

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes.

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub-nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

There are two types of Nodes

- **Device Node:** Device nodes are physical computing resources with processing memory and services to execute software, such as typical computers or mobile phones.
- **Execution Environment Node:** An execution environment node (EEN) is a software computing resource that runs.



**Figure 3.6.7 Deployment Diagram**

## 3.7 Database Design

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

### **3.7.1 ENTITY RELATION(ER-DIAGRAM)**

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attribute also called an entity-relationship model, a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data, an object or concept about which data is stored. A relationship is how the data is shared between entities. Several types of attributes occur in the ER model:

- Simple Attribute
- Composite Attributes
- Single-Valued Attributes

- Multivalued Attributes
- Stored Attributes
- Derived Attributes

### **3.7.2 NORMALIZATION**

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this system for the following reasons.

- To structure the data so that there is no repetition of data, this helps in saving.
- To permit simple retrieval of data in response to query and report request.
- To simplify the maintenance of the data through updates, insertions, deletions.
- To reduce the need to restructure or reorganize data which new application requirements arise?

As the name implies, it denoted putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. This includes:

- Normalize the data.
- Choose proper names for the tables and columns.
- Choose the proper name for the data.

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updating, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relation.

**Insertion anomaly:** Inability to add data to the database due to absence of other data.

**Deletion anomaly:** Unintended loss of data due to deletion of other data.

**Update anomaly:** Data inconsistency resulting from data redundancy and partial update

➤ **Normal Forms:** These are the rules for structuring relations that eliminate anomalies.

**First Normal Form:** A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

**Second Normal Form:**

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

- Primary key is a not a composite primary key
- No non key attributes are present
- Every non-key attribute is fully functionally dependent on full set of primary key.

**Third Normal Form:** A relation is said to be in third normal form if their exists no transitive dependencies. Transitive Dependency

If two non-key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

### 3.7.3 SCHEMA

The description of a database is called the database schema, which is specified during the database design and is not expected to change frequently. Most data module has certain.

Conventions for displaying for the schema's as diagram. A displayed schema is called a schema diagram. Figure below show the schema diagram for our database:

A Data Dictionary is a collection of metadata that is data about data. In addition to storing catalog information about schemas and constraints, the data dictionary stores other information, such as design decisions, usage standards,

# **CHAPTER 04**

## **IMPLEMENTATION**

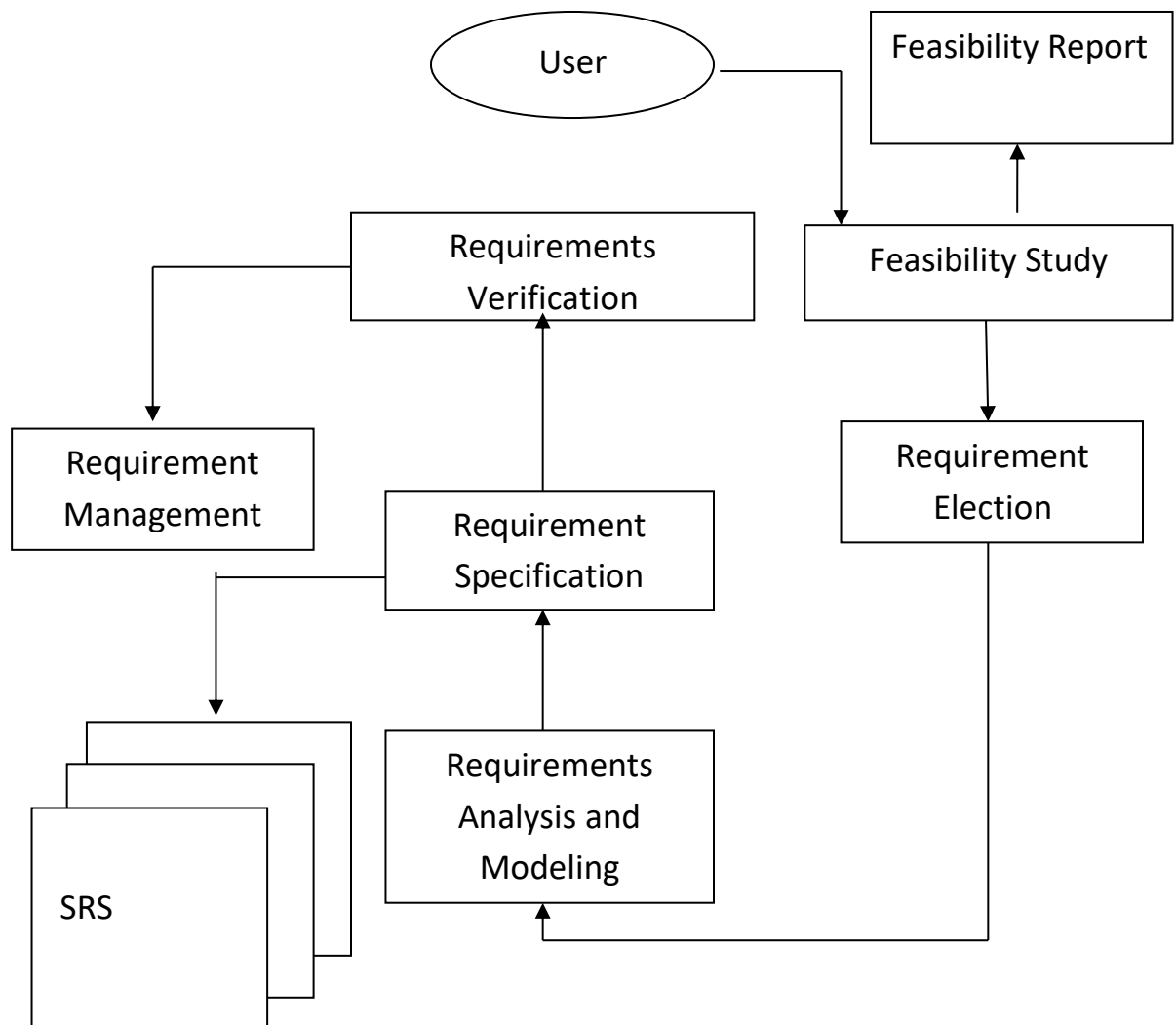
## **4. SYSTEM IMPLEMENTATION**

### **4.1 Software Requirement Specifications**

A software requirement specification is developed as consequence of analysis. Review is essential to ensure that the developer and customer have the same perception. Software requirements specification (SRS) is the starting point of the software development activity. The software requirements specification is produced at culmination of the analysis task. The introduction of the software requirement specification states the goals and objectives of the software, describing it in the context of the computer based system.

The software requirements specification includes an information description, functional description, behavioral description. There are several models for such processing, each describing approaches to a variety of tasks or activities that take place during the process. Some people consider a life cycle model a more general term and a software development process a more specific term. For example, there are meant specific software development processes that 'fit' the spiral life cycle model. It aims to be the standard that defines all the tasks required for developing and maintaining software. The introduction of the software requirement specification states the goals and objectives of the software, describing it in the context of the computer based system.

The purpose of this document is present the software requirement in a precise and easily understood manner. This document provides the functional, performance design and verification requirement of the software to be developed this is the only document that describes the requirement of the system. This is meant for use by the developers and will also be the basis for validating the final delivered system. Figure 4.1 describes software requirement analysis in a diagrammatic manner which is easily understandable.



**Fig. 4.1 Software Requirement Analysis**

## **4.2 Functional Requirements**

Functional requirements describe what the system should do, i.e. the services provided for the users and for the other systems.



## **4.3 Non-Functional Requirement**

Non-Functional requirement are constraints that must be adhered during development. They limit what resource can be used and set bounds on aspects of the software's quality. One of the most important things about non-functional requirement is to be making them verifiable. The verification is normally done by measuring various aspects of the system and seeing if the measurement confirm to the requirements.

### **4.3.1 PERFORMANCE REQUIREMENTS**

Some Performance requirements identified is listed below:

1. The database shall be able to accommodate a minimum of 10,000 records of students.
2. The software shall support use of multiple users at a time.
3. There are no other specific performance requirements that will affect development.

### **4.3.2 SAFETY REQUIREMENTS**

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.

### **4.3.3 SECURITY REQUIREMENTS**

Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below. Keep specific log or history data sets

1. Assign certain functions to different modules
2. Restrict communications between some areas of the program.
3. Check data integrity for critical variables.

4. Later version of the software will incorporate encryption techniques in the user/license authentication process.

In non-functional requirements the following are the things that come under .They are as follows:

- 1) **Reusability:** As we developed the application in java, the application can be re-used for any one without having any restrictions in its usage. Hence it is re-Usable.
- 2) **Portability:** As the application is designed with java as programming language, we know java can be run on any operating system. Hence the application is portable to run on any operating system.
- 3) **Extensibility:** The application can be extended at any level if the user wish to extend that in future this is done because java is a open source medium which doesn't have any time limits for expiry or renewal. The same application can be re-used by any one for further extent without any time restrictions.

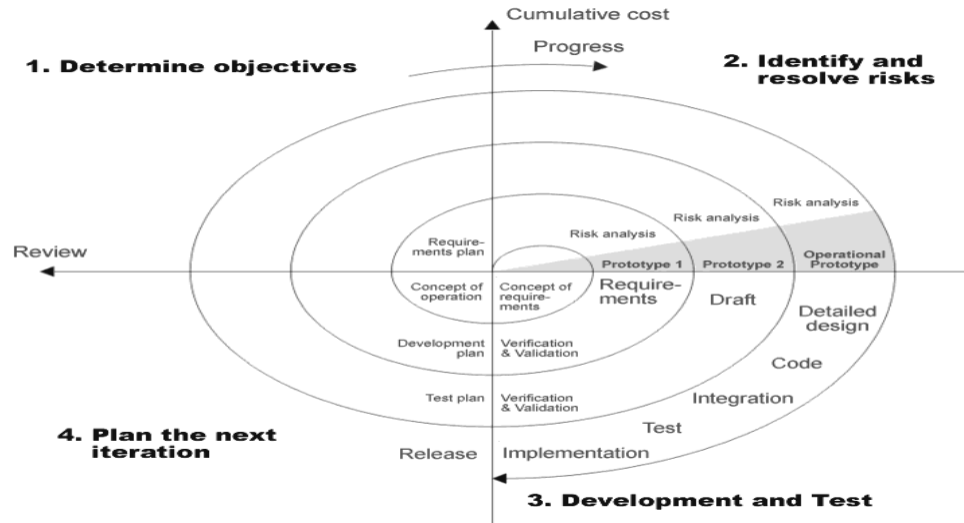
## 4.4 Model Used:Spiral Model

### 4.4.1 Introduction

The spiral model combines the idea of iterative development (prototyping) with the systematic, controlled aspects of the waterfall model. It allows for incremental releases of the product, or incremental refinement through each time around the spiral as shown below in figure 4.2

The spiral model also explicitly includes [risk management](#) within [software development](#). Identifying major risks, both technical and managerial, and determining how to lessen the risk helps keep [software development process](#) under control.

The spiral model is based on continuous refinement of key products for requirements definition and [analysis](#), [system](#) and [software design](#), and [implementation](#) (the code). At each iteration around the cycle, the products are extensions of an earlier product.



**Fig.4.5.Spiral Model**

Documents are produced when they are required, and the content reflects the information necessary at that point in the process. All documents will not be created at the beginning of the process, nor all at the end (hopefully). Like the product they define, the documents are works in progress. Starting at the center, each turn around the spiral goes through several task regions determine the objectives, alternatives, and constraints on the new iteration.

- Evaluate alternatives and identify and resolve risk issues.
  - Develop and verify the product for this iteration.
  - Plan the next iteration.
- Each phase is split in 4 quadrants, which are:
- Determine objective.
  - Identify and resolve risk.
  - Develop next level product.
  - Review and planning.

### 4.4.2 Advantages

The primary advantage of the spiral model is that its range of options accommodates the good features of existing software process models, while its risk-driven approach avoids many of their difficulties. In appropriate situations, the spiral model becomes equivalent to one of the existing process models. In other situations, it provides guidance on the best mix of existing approaches to a given project; for example, its application to the TRW-SPS provided a risk-driven mix of specifying, prototyping, and evolutionary development.

The primary conditions under which the spiral model becomes equivalent to other main process models are summarized as follows:

- If a project has a low risk in such areas as getting the wrong user interface or not meeting strength performance requirements, and if it has a high risk in budget and schedule predictability and control, then these risk considerations drive the spiral model into equivalence to the waterfall model.
- If a software product's requirements are very stable (implying a low risk of expensive design and code breakage due to requirements changes during development), and if the presence of errors in the software product constitutes a high risk to the mission it serves, then these risk considerations drive the spiral model to resemble the two-leg model of precise specification and formal deductive program development.
- If a project has a low risk in such areas as losing budget and schedule predictability and control, encountering large-system integration problems, or coping with information sclerosis, and if it has a high risk in such areas as getting the wrong user interface or user decision support requirements, then these risk considerations drive the spiral model into an equivalence to the evolutionary development model.
- If automated software generation capabilities are available, then the spiral model accommodates them either as options for rapid prototyping or for application of the transform model, depending on the risk considerations involved.

If the high-risk elements of a project involve a mix of the risk items listed above, then the spiral approach will reflect an appropriate mix of the process models above (as exemplified in

the TRW-SPS application). In doing so, its risk avoidance features will generally avoid the difficulties of the other models.

#### 4.5 Hardware Specifications

Description	Minimum	Recommended
System type	P IV 2.8 GHz and above	Core Processors
RAM	256 MB	1 GB MB
Cache	256 MB	1 GB
Storage	40 GB	500 GB
Disk Drive	Floppy or CD-RW (back up purpose)	CD-RW or DAT(back up purpose)
Display	15" VGA	17" LED
Output media	DMP Printer	Laser Printer

**Table 4.6. Hardware Specification**

#### 4.6 Software Specifications

- Operating system : Windows 7 Ultimate.
- Coding Language : Python, PyTorch, OPENCV
- Front-End : Python.
- Technologies : ML & Deep Learning
- Back end Dataset : Collected From CCTV Footage

## 4.7 Theoretical Background

The following languages are used built the required software. They are:

### 4.7.1 Front End Implementation

#### Python Introduction

**Python** is a general purpose, dynamic, high level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development. Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development. It supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles. Python is not intended to work in a particular area, such as web programming. That is why it is known as multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc. We don't need to use data types to declare variable because it is dynamically typed so we can write `a=10` to assign an integer value in an integer variable. It makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

#### Python Applications

Python is known for its general purpose nature that makes it applicable in almost each domain of software development. Python as a whole can be used in any sphere of development. Here, we are specifying applications areas where python can be applied.

- **Web Applications:** We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, BeautifulSoup, Feedparser etc. It also provides Frameworks such as Django, Pyramid, Flask etc to design and develop web based applications. Some important developments are: PythonWikiEngines, Pycoco, PythonBlogSoftware etc. The useful library and package are SciPy, Pandas, IPython etc. SciPy is group of packages of engineering.

- **Desktop GUI Applications:** Python provides Tk GUI library to develop user interface in python based application. Some other useful toolkits wxWidgets, Kivy, pyqt that are useable on several platforms. The Kivy is popular for writing multitouch applications.
- **Software Development:** Python is helpful for software development process. It works as a support language and can be used for build control and management, testing etc.
- **Scientific and Numeric:** Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, IPython etc. SciPy is group of packages of engineering, science and mathematics.
- **Business Application:** Python is used to build Bussiness applications like ERP and e-commerce systems. Tryton is a high level application platform.
- **Console Based Application:** We can use Python to develop console based applications. For example: IPython.
- **Audio or Video based Applications:** Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of real applications are: TimPlayer, cplay etc.\
- **3D CAD Applications:** To create CAD application Fandango is a real application which provides full features of CAD.
- **Enterprise Applications:** Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications are: OpenErp, Tryton, Picalo etc.
- **Applications for Images:** Using Python several application can be developed for image. Applications developed are: VPython, Gogh, imgSeek etc. Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development. It supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles.

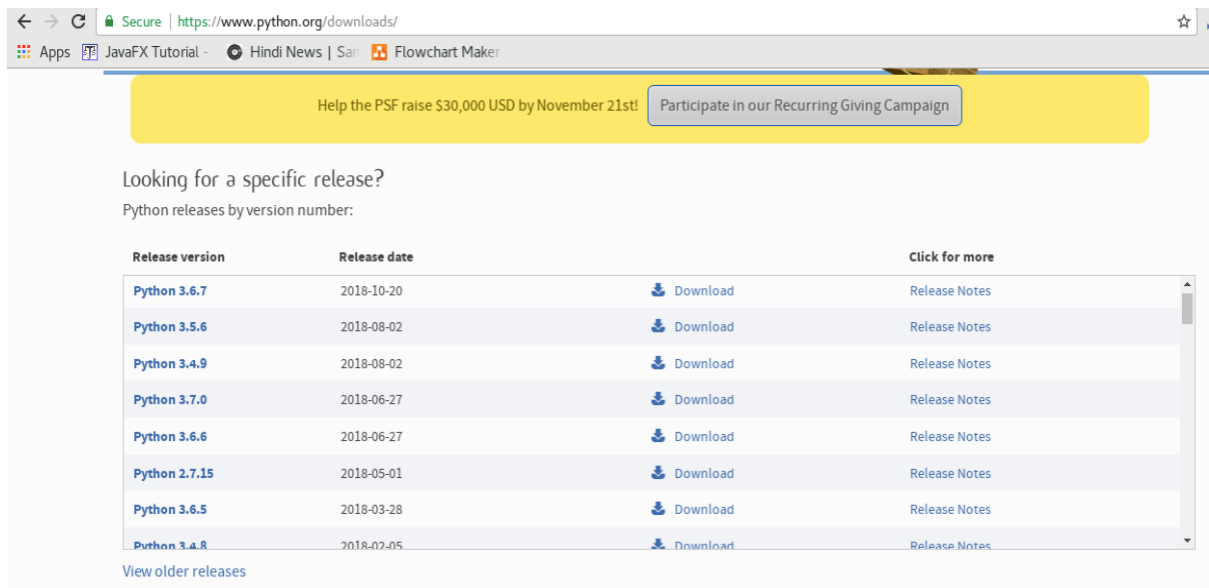
There are several such applications which can be developed using Python

### **How to Install Python (Environment Set-up)**

In this section of the tutorial, we will discuss the installation of python on various operating systems.

## Installation on Windows:

Visit the link <https://www.python.org/downloads/> to download the latest release of Python. In this process, we will install Python 3.6.7 on our Windows operating system.



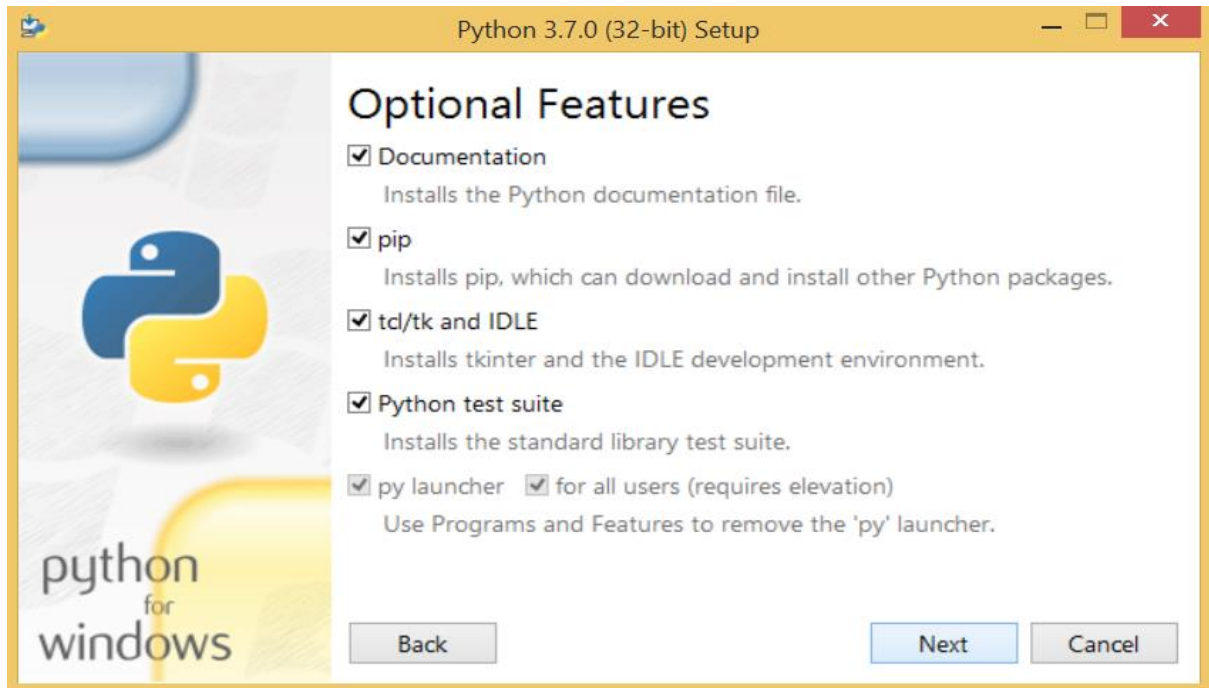
Double-click the executable file which is downloaded; the following window will open. Select Customize installation and proceed.



**Fig. 4.1 Python Installation**

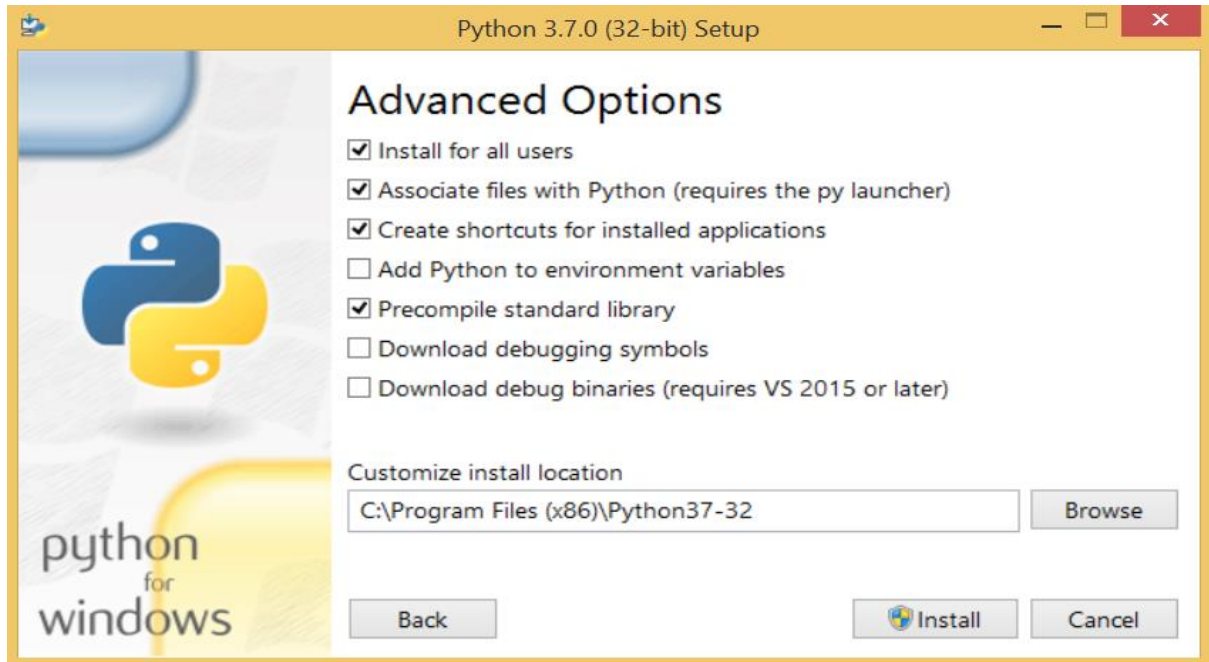


The following window shows all the optional features. All the features need to be installed and are checked by default; we need to click next to continue.

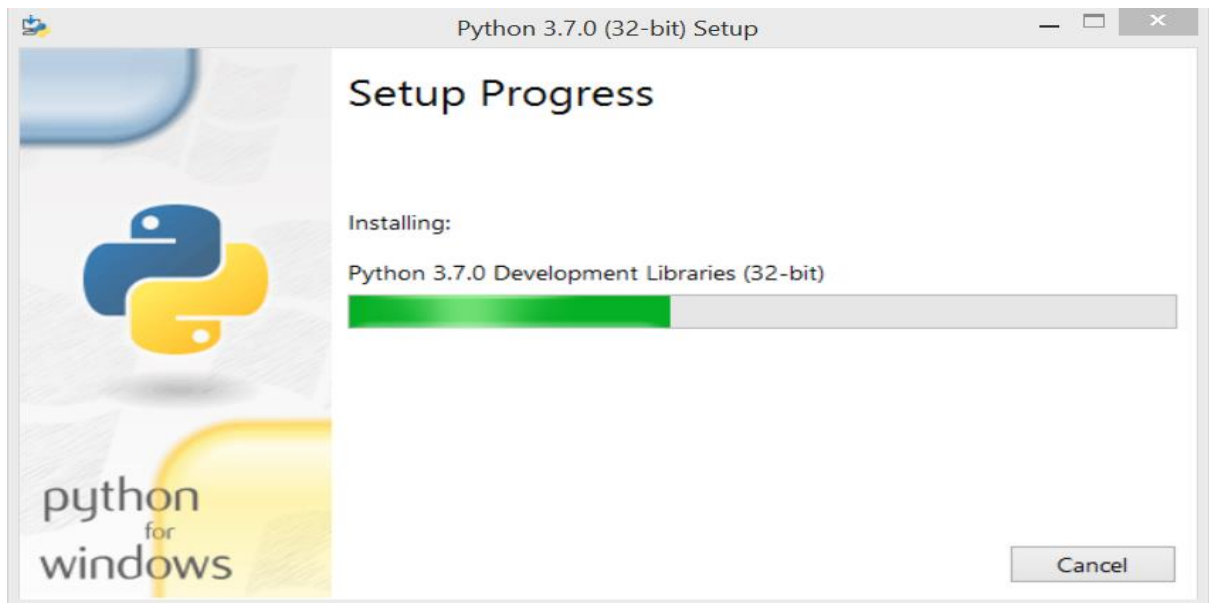


**Fig. 4.2 Python Setup**

The following window shows a list of advanced options. Check all the options which you want to install and click next. Here, we must notice that the first check-box (install for all users) must be checked.

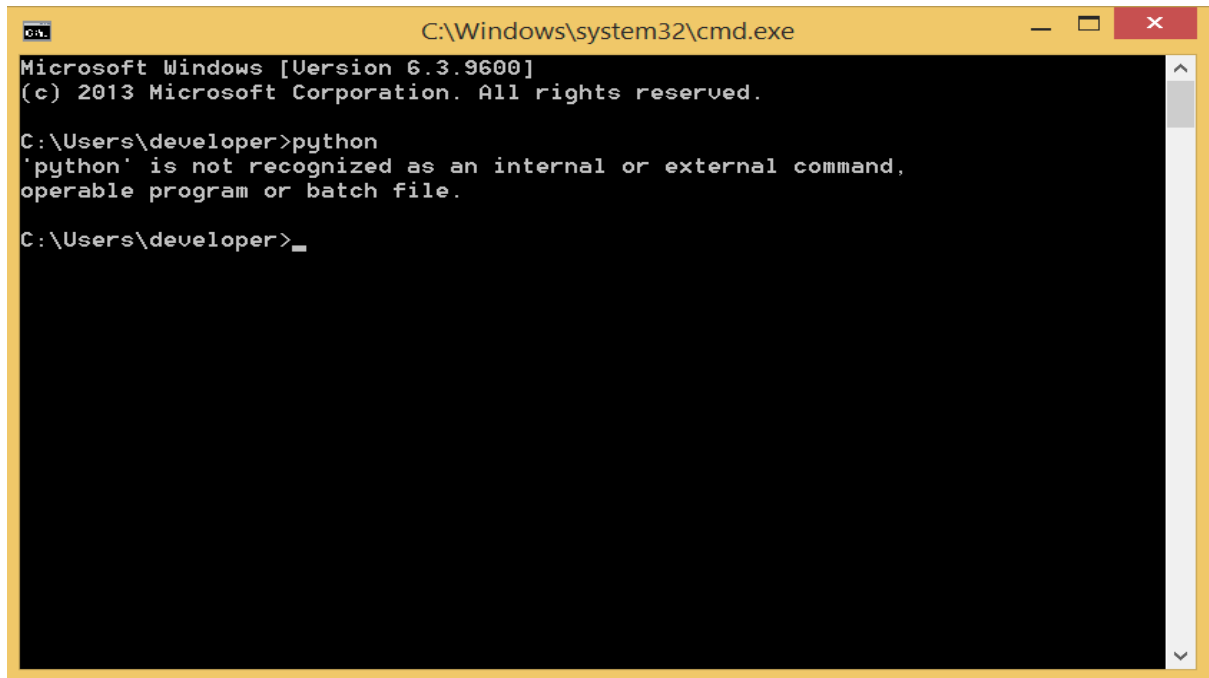


Now, we are ready to install python-3.6.7. Let's install it.



**Fig. 4.3 Python Setup Progress**

Now, try to run python on the command prompt. Type the command **python** in case of python2 or python3 in case of **python3**. It will show an error as given in the below image. It is because we haven't set the path.

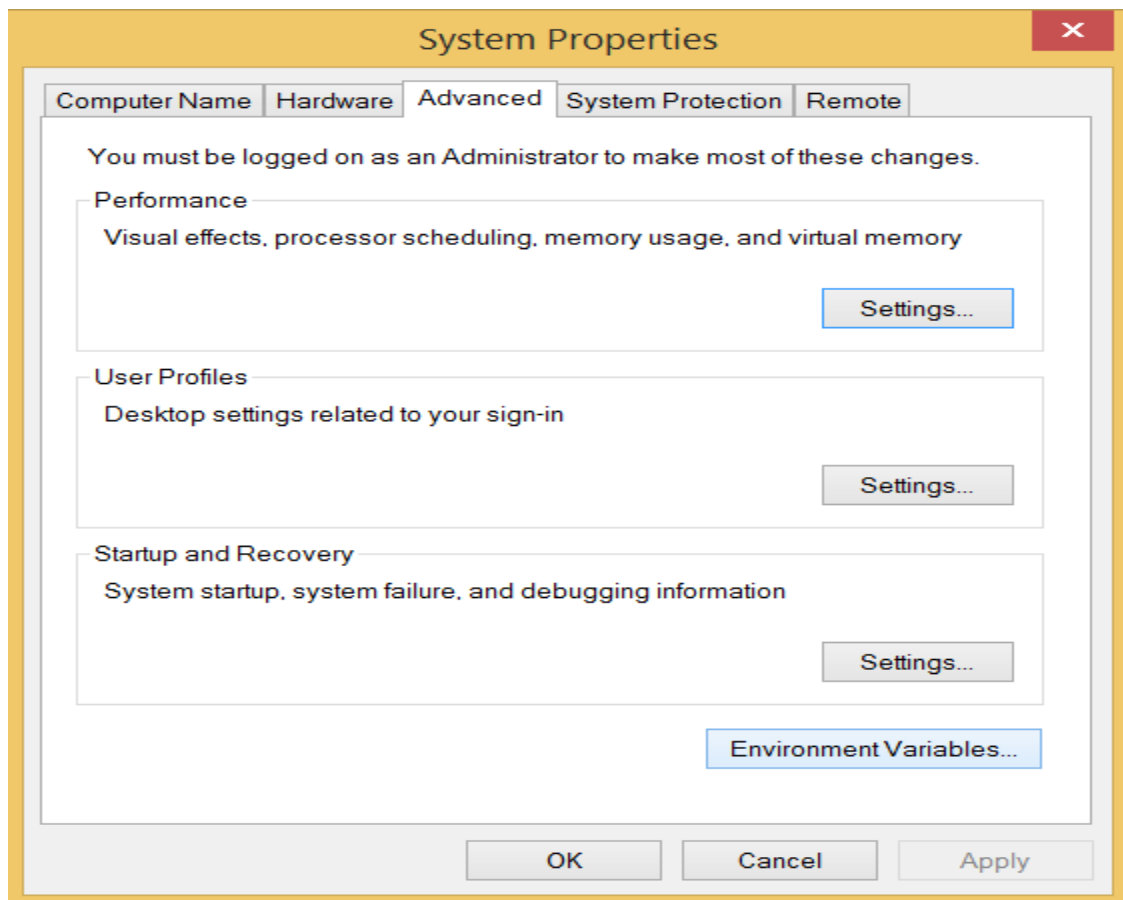
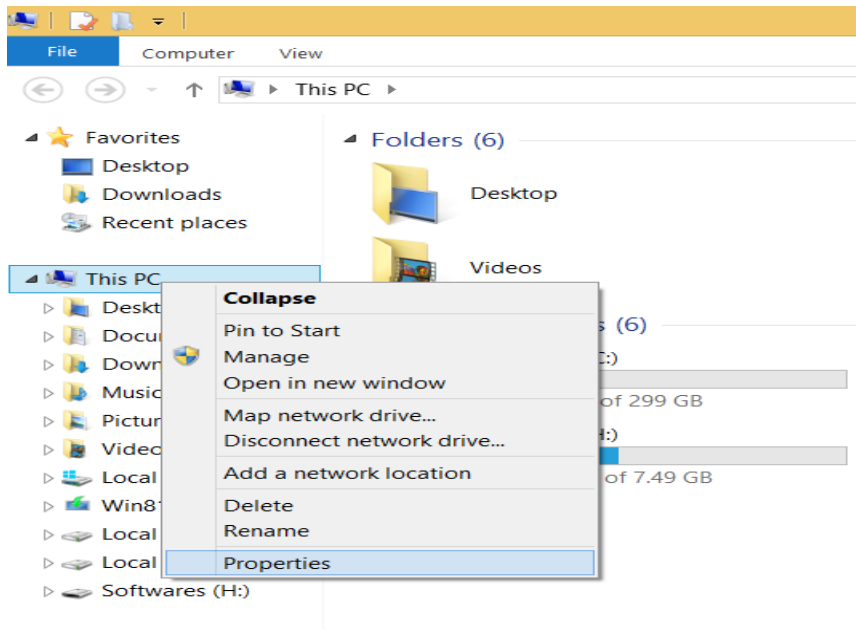


```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

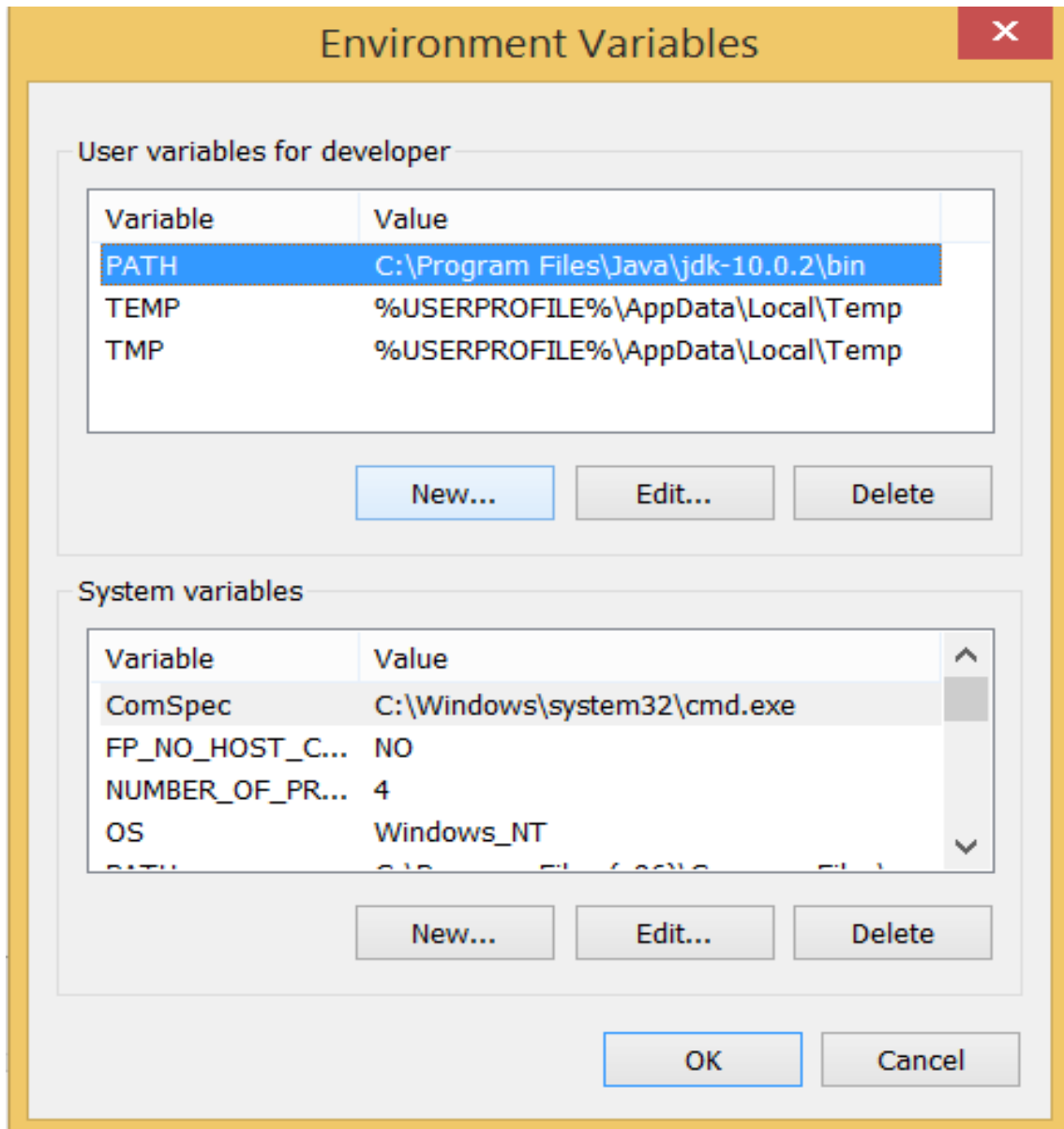
C:\Users\developer>python
'python' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\developer>
```

To set the path of python, we need to right click on "my computer" and go to Properties → Advanced → Environment Variables.

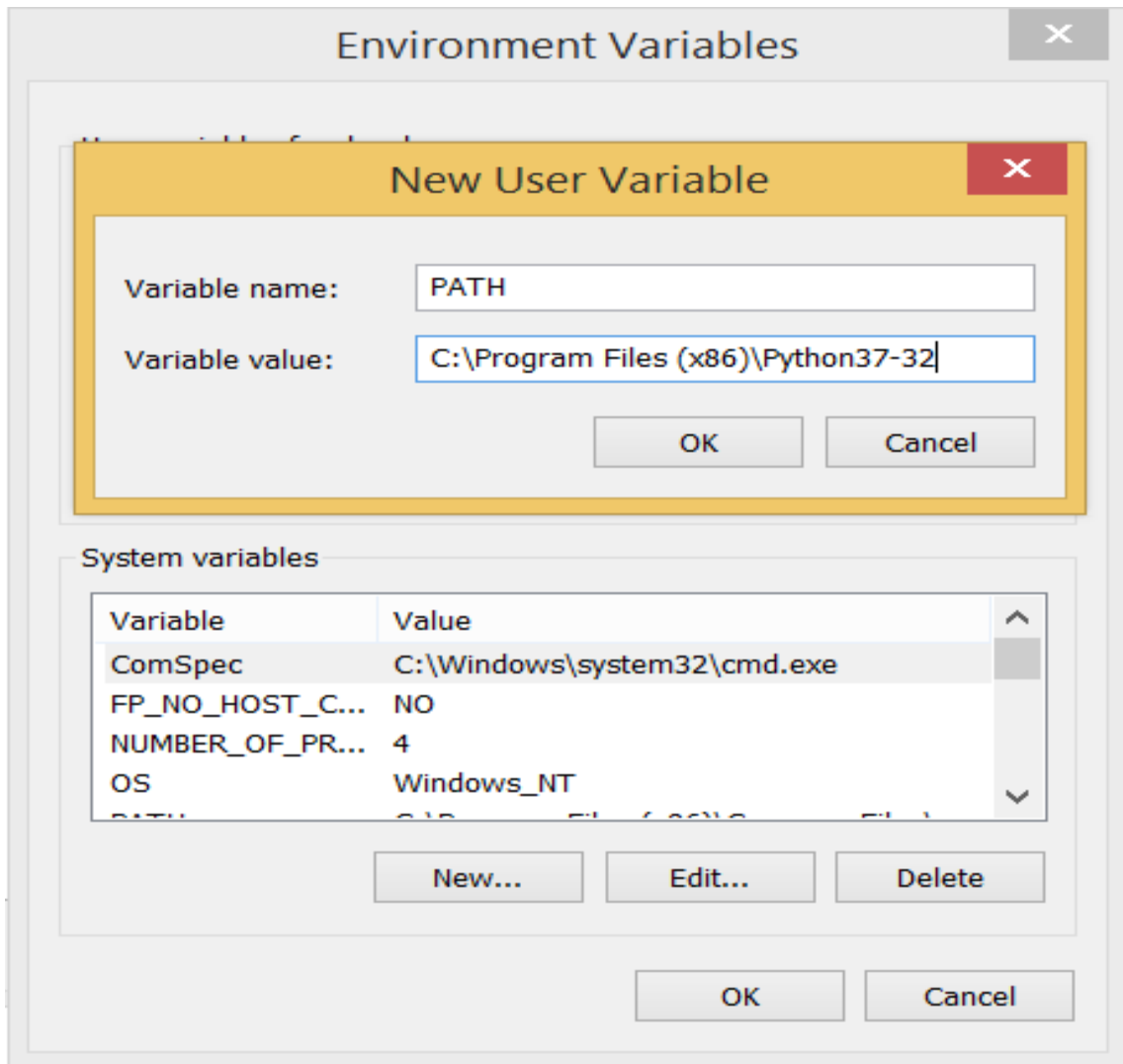


Add the new path variable in the user variable section.



**Fig. 4.4 Python Path Setup**

Type **PATH** as the variable name and set the path to the installation directory of the python shown in the below image.



Now, the path is set, we are ready to run python on our local system. Restart CMD, and type **python** again. It will open the python interpreter shell where we can execute the python statements.

## First Python Program

In this Section, we will discuss the basic syntax of python by using which, we will run a simple program to print hello world on the console.

Python provides us the two ways to run a program:

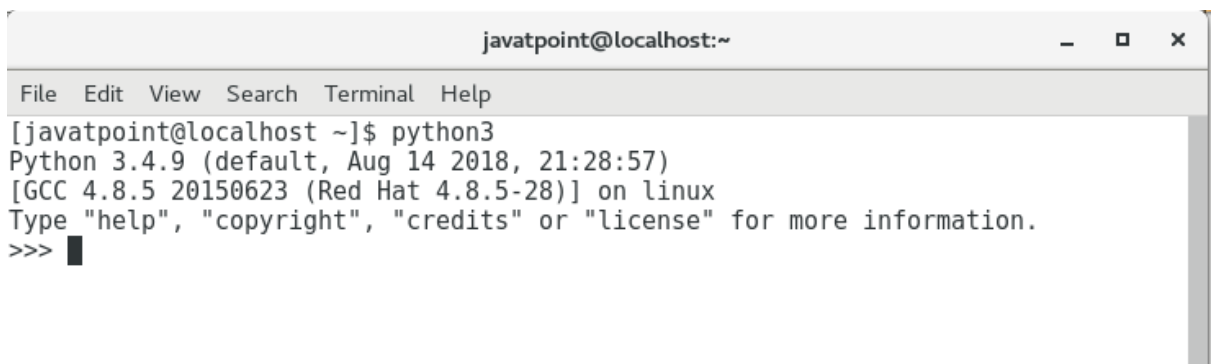
- Using Interactive interpreter prompt
- Using a script file

Let's discuss each one of them in detail.

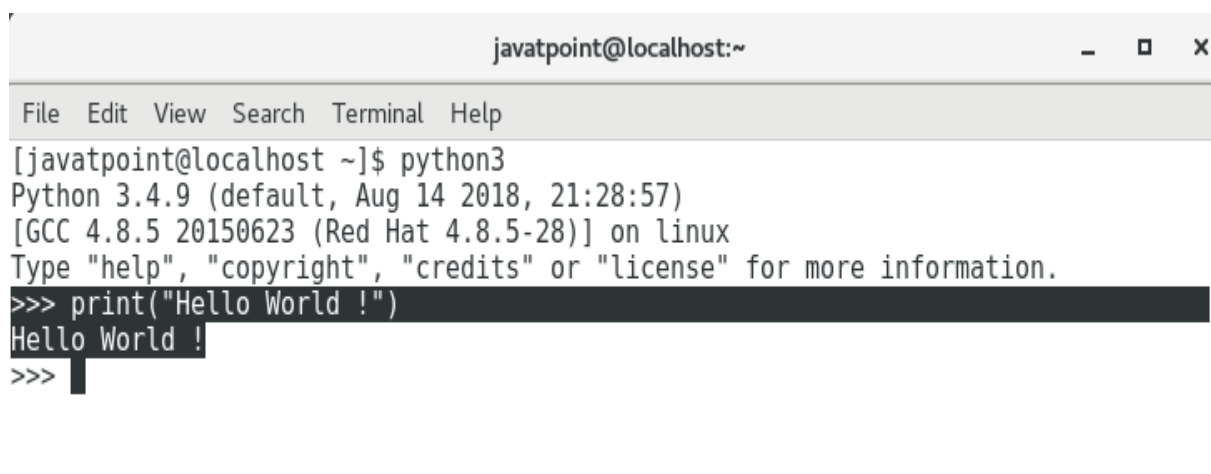
## Interactive interpreter prompt

Python provides us the feature to execute the python statement one by one at the interactive prompt. It is preferable in the case where we are concerned about the output of each line of our python program. To open the interactive mode, open the terminal (or command prompt) and type python (python3 in case if you have python2 and python3 both installed on your system).

It will open the following prompt where we can execute the python statement and check their impact on the console.

A screenshot of a terminal window titled 'javatpoint@localhost:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal content shows the command '[javatpoint@localhost ~]\$ python3' being executed, followed by the Python version and GCC information: 'Python 3.4.9 (default, Aug 14 2018, 21:28:57) [GCC 4.8.5 20150623 (Red Hat 4.8.5-28)] on linux'. It then prompts the user to 'Type "help", "copyright", "credits" or "license" for more information.' and shows the interactive prompt '>>>' with a cursor.

Let's run a python statement to print the traditional hello world on the console. Python3 provides print() function to print some message on the console. We can pass the message as a string into this function. Consider the following image.

A screenshot of a terminal window titled 'javatpoint@localhost:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal content shows the command '[javatpoint@localhost ~]\$ python3' being executed, followed by the Python version and GCC information: 'Python 3.4.9 (default, Aug 14 2018, 21:28:57) [GCC 4.8.5 20150623 (Red Hat 4.8.5-28)] on linux'. It then prompts the user to 'Type "help", "copyright", "credits" or "license" for more information.' and shows the interactive prompt '>>>'. The user enters the command 'print("Hello World !")', and the output 'Hello World !' is printed on the next line. The prompt '>>>' is shown again on the following line.

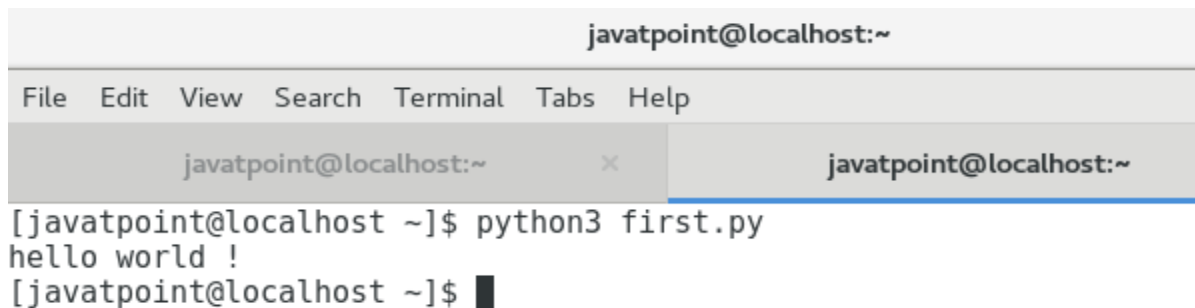
Here, we get the message "**Hello World !**" printed on the console.

## Using a script file

Interpreter prompt is good to run the individual statements of the code. However, we cannot write the code every-time on the terminal. We need to write our code into a file which can be executed later. For this purpose, open an editor like notepad, create a file named first.py (python used .py extension) and write the following code in it.

- Print ("hello world"); #here, we have used print() function to print the message on the console. To run this file named as first.py, we need to run the following command on the terminal.

**\$ python3 first.py**



The screenshot shows a terminal window titled 'javatpoint@localhost:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', 'Tabs', and 'Help'. Below the menu bar, there are two tabs, both labeled 'javatpoint@localhost:~'. The active tab shows the following text:

```
[javatpoint@localhost ~]$ python3 first.py
hello world !
[javatpoint@localhost ~]$
```

Hence, we get our output as the message **Hello World !** is printed on the console.

## Get Started with PyCharm

In our first program, we have used gedit on our CentOS as an editor. On Windows, we have an alternative like notepad or notepad++ to edit the code. However, these editors are not used as IDE for python since they are unable to show the syntax related suggestions. JetBrains provides the most popular and a widely used cross-platform IDE **PyCharm** to run the python programs.

## PyCharm installation

As we have already stated, PyCharm is a cross-platform IDE, and hence it can be installed on a variety of the operating systems. In this section of the tutorial, we will cover the installation process of PyCharm on Windows, MacOS, CentOS, and Ubuntu.



## Windows

Installing PyCharm on Windows is very simple. To install PyCharm on Windows operating system, visit the link <https://www.jetbrains.com/pycharm/download/download-thanks.html?platform=windows> to download the executable installer. Double click the installer (.exe) file and install PyCharm by clicking next at each step.

## Anaconda

Some tutorials install packages with conda instead of pip. So what is conda?



Anaconda is a distribution of Python (and R). It is free and open-source and makes package management and deployment simpler. Keep reading to see how. It is the standard platform for python data Science and open-source machine learning. Anaconda is used by data scientists, IT professionals and business leaders.

### Benefits of Using Python Anaconda

Why should you use Anaconda for your project? Well, it does have the following benefits:

- It is free and open-source.
- It has more than 1500 Python/R data science packages.
- Anaconda simplifies package management and deployment.
- It has tools to easily collect data from sources using machine learning and AI.
- It creates an environment that is easily manageable for deploying any project
- Anaconda is the industry standard for developing, testing and training on a single machineIt has good community support- you can ask your questions there.

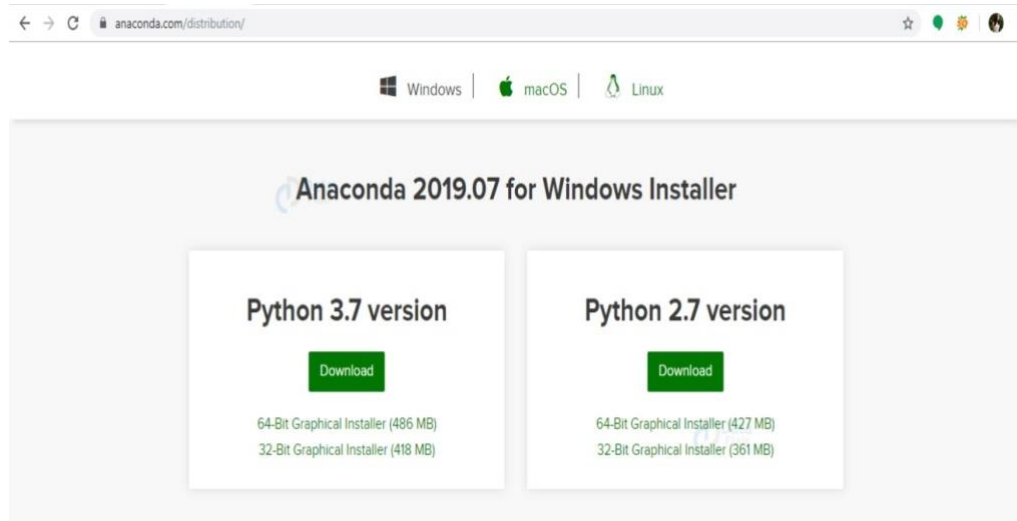
What you get

- Download more than 1500 Python/R data science packages
- Manage libraries, dependencies, and environments with conda
- Build and train ML and deep learning models with scikit-learn, TensorFlow and Theano
- Use Dask, NumPy, Pandas and Numba to analyze data scalably and fast
- Perform visualization with Matplotlib, Bokeh, Datashader, and Holoviews

## Python Anaconda Installation

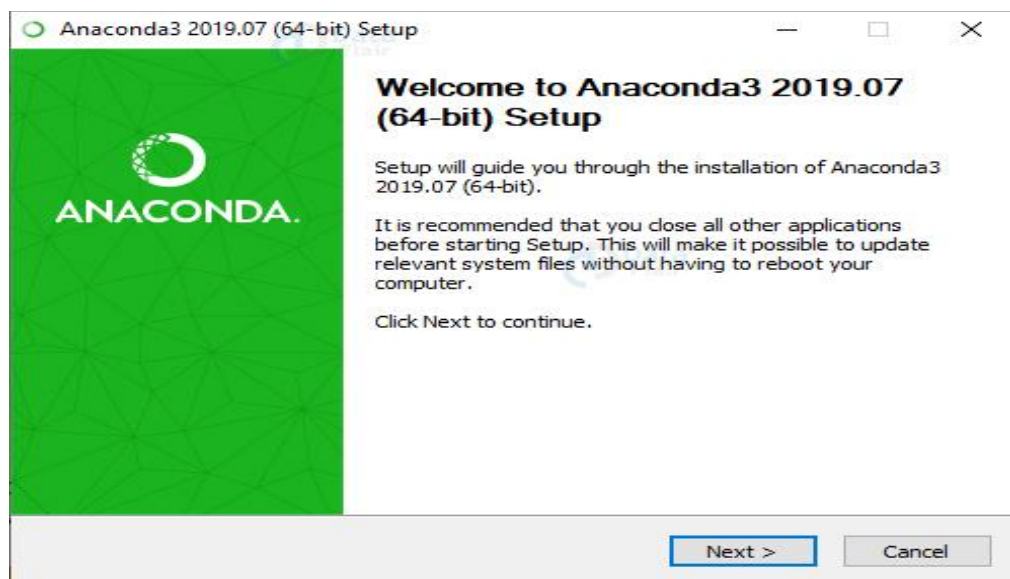
Next in the Python anaconda tutorial is its installation. The latest version of Anaconda at the time of writing is 2019.07. Follow these steps to download and install Anaconda on your machine:

- Go to this link and download Anaconda for Windows, Mac, or Linux: – Download anaconda



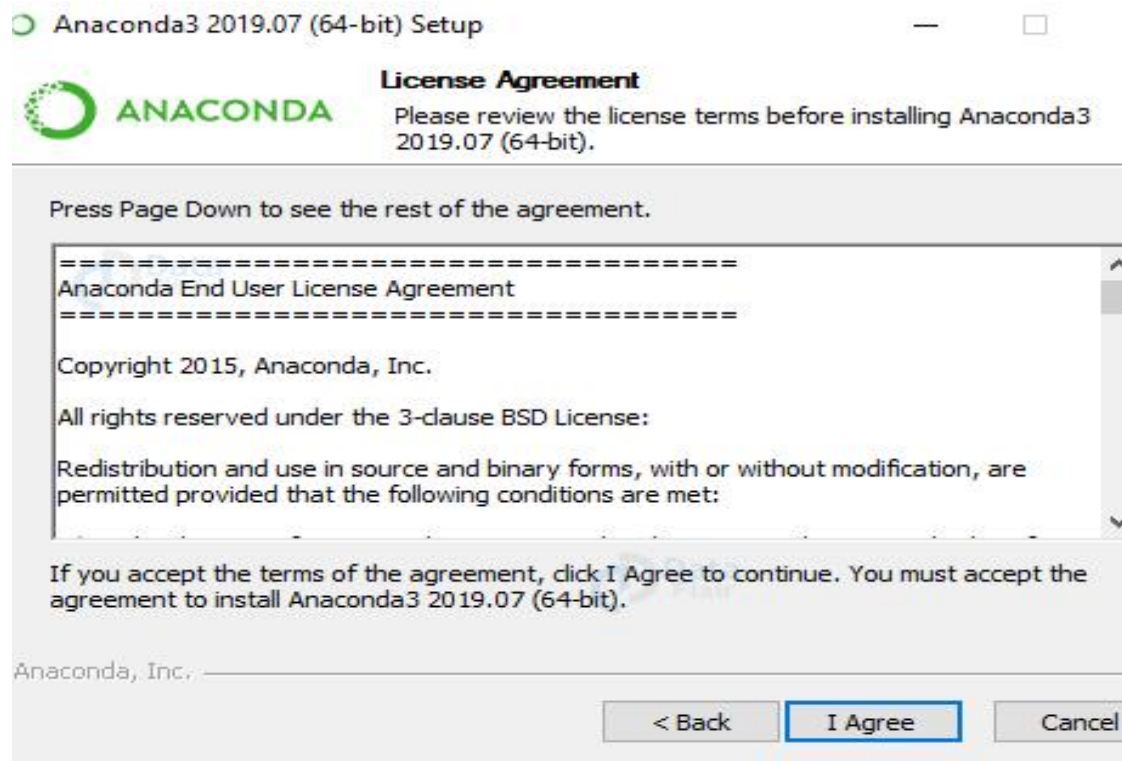
You can download the installer for Python 3.7 or for Python 2.7 (at the time of writing). And you can download it for a 32-bit or 64-bit machine.

- Click on the downloaded .exe to open it. This is the Anaconda setup. Click next.

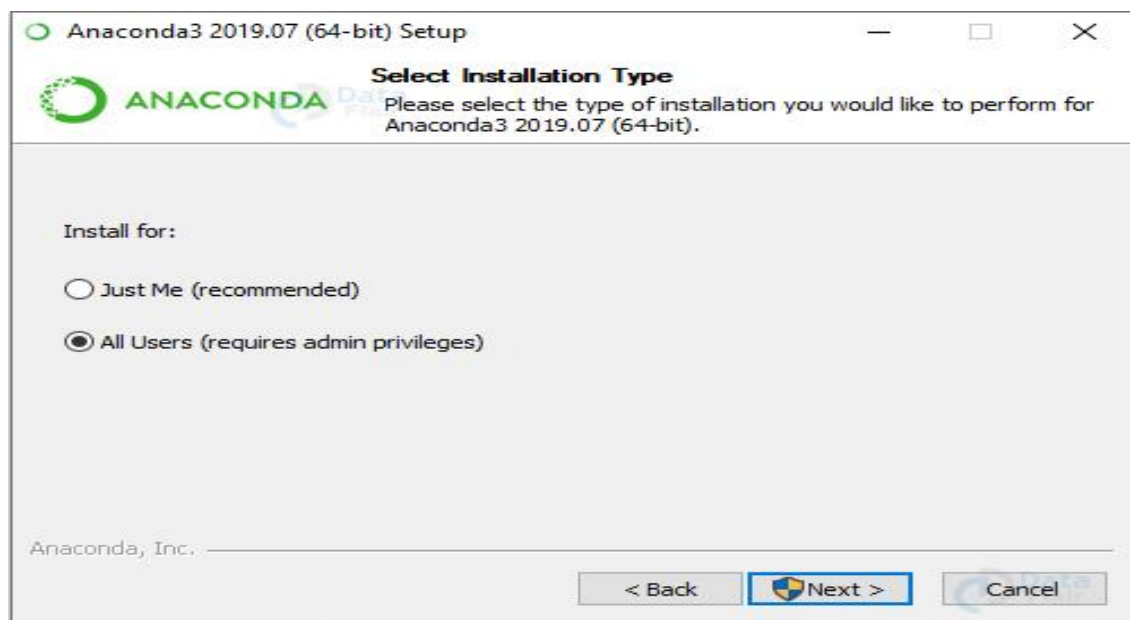


**Fig: 4.5 Anaconda Setup**

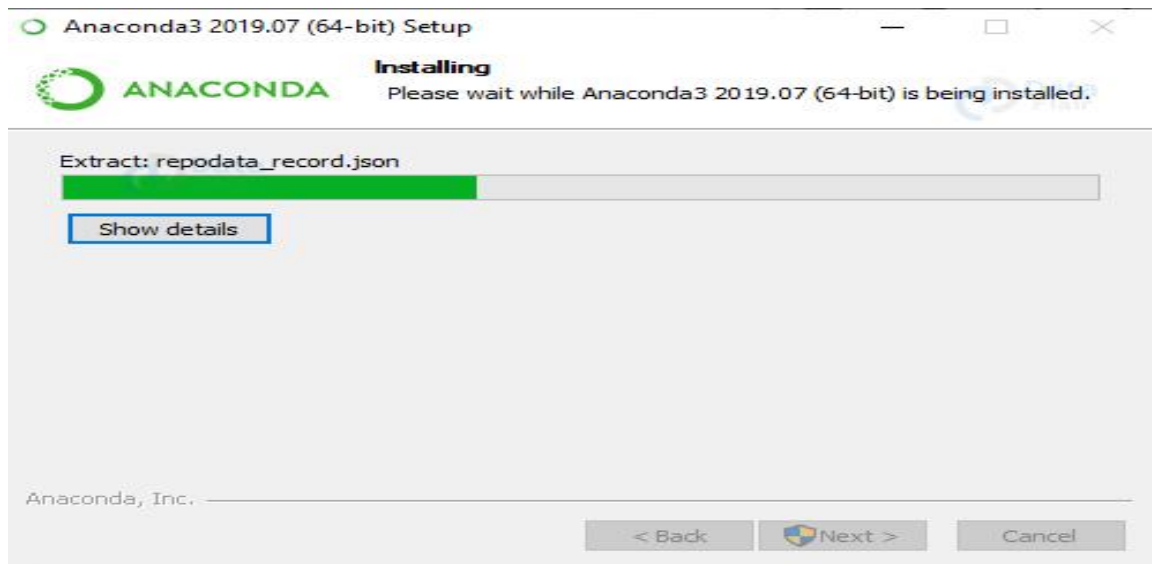
- Now, you'll see the license agreement. Click on 'I Agree'.



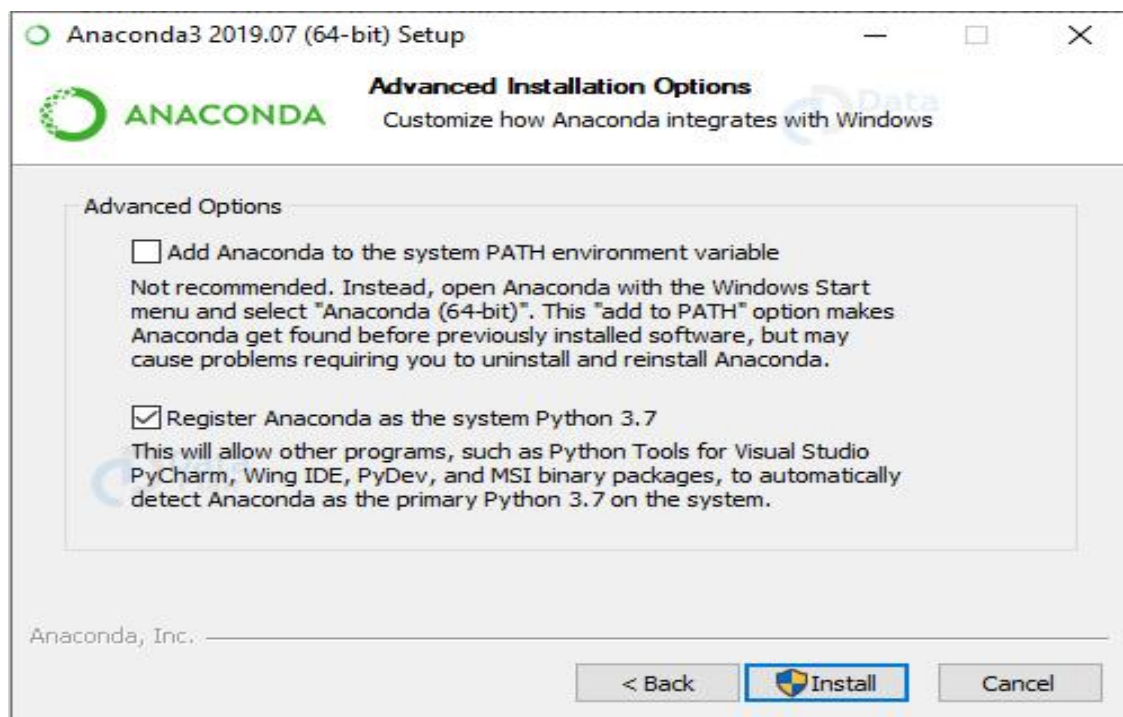
- You can install it for all users or just for yourself. If you want to install it for all users, you need administrator privileges.



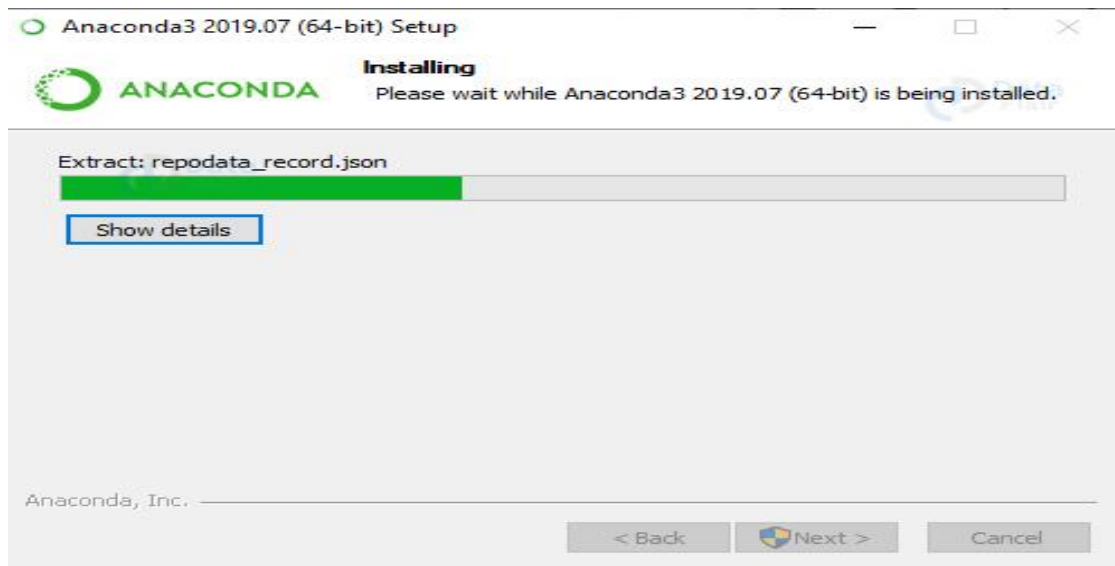
- Choose where you want to install it. Here, you can see the available space and how much you need.



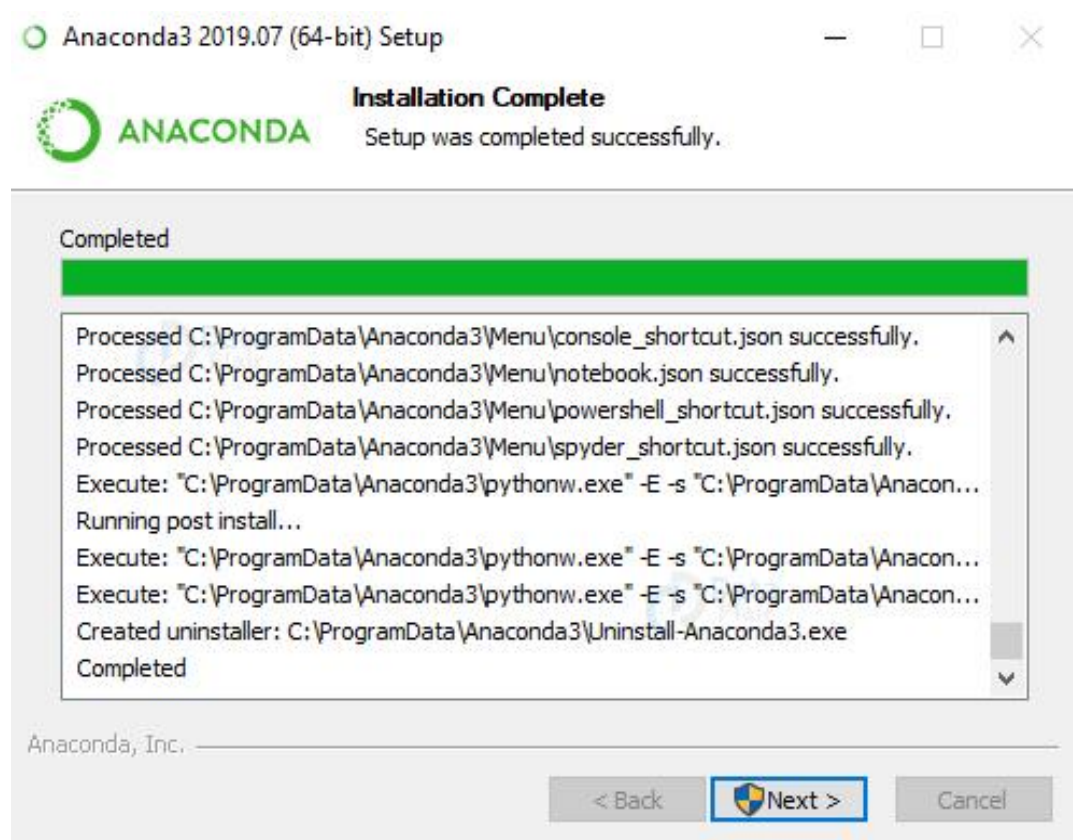
- Now, you'll get some advanced options. You can add Anaconda to your system's PATH environment variable, and register it as the primary system Python 3.7. If you add it to PATH, it will be found before any other installation. Click on 'Install'.



- It will unpack some packages and extract some files on your machine. This will take a few minutes.

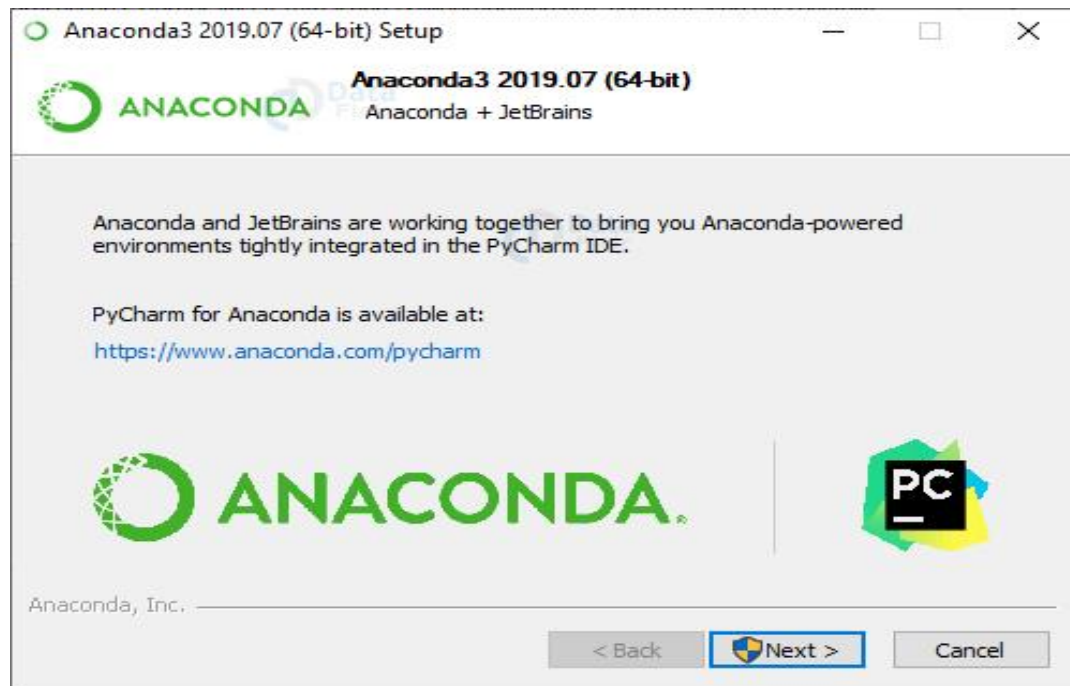


- The installation is complete. Click Next.



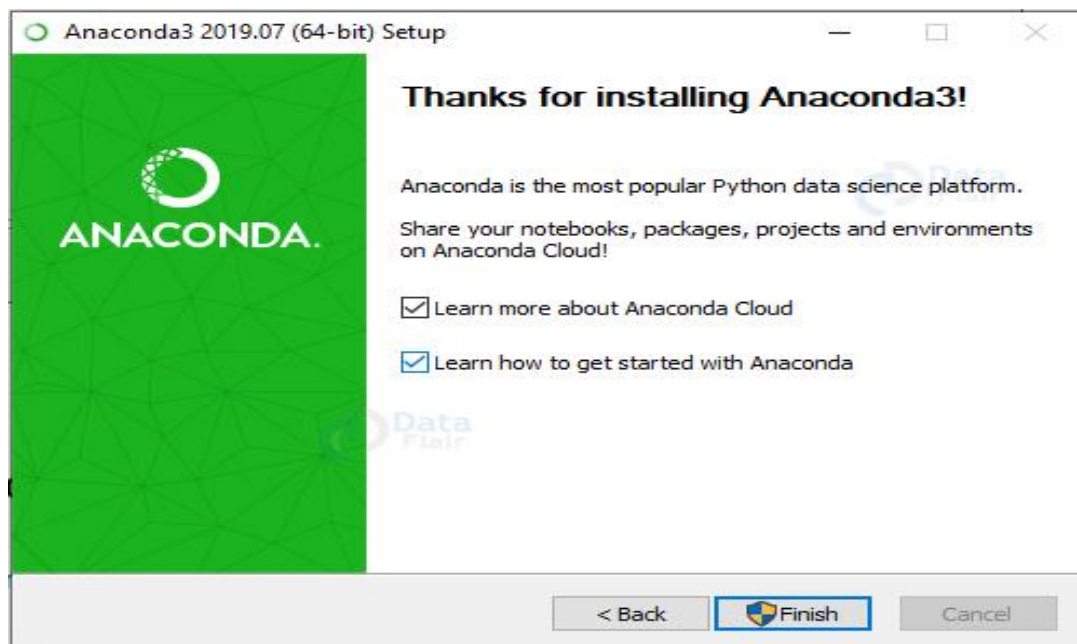


- This screen will inform you about PyCharm. Click Next.

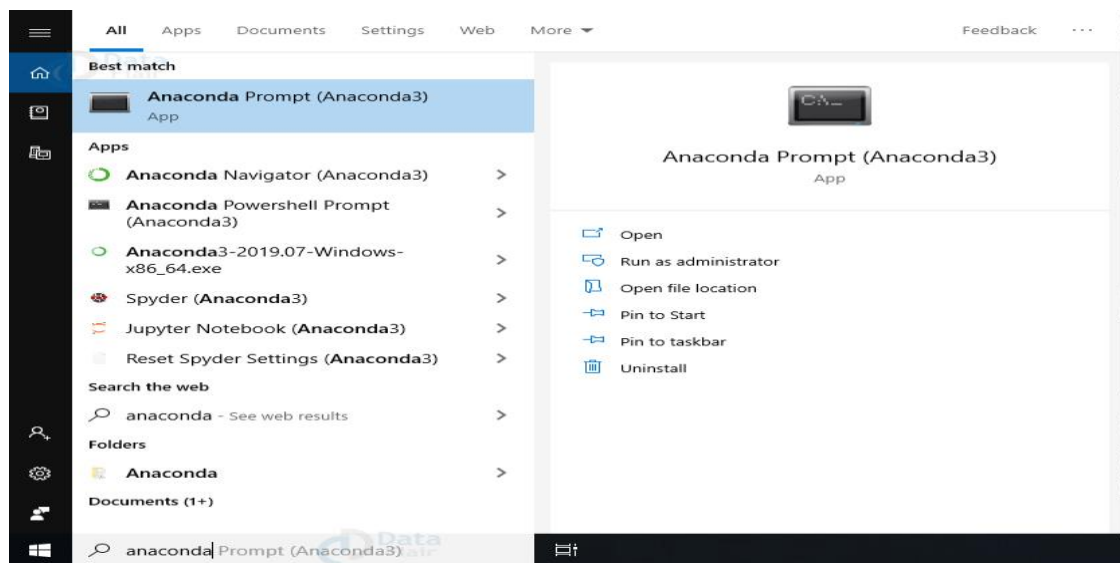


**Fig: 4.6 Pycharm IDE**

- The installation is complete. You can choose to get more information about Anaconda cloud and how to get started with Anaconda.



- If you search for Anaconda now, you will see the following options:



**Fig: 4.7 Anaconda Prompt**

- The Anaconda prompt
- Anaconda Navigator
- Anaconda Powershell prompt
- Spyder IDEand
- Jupyter Notebook

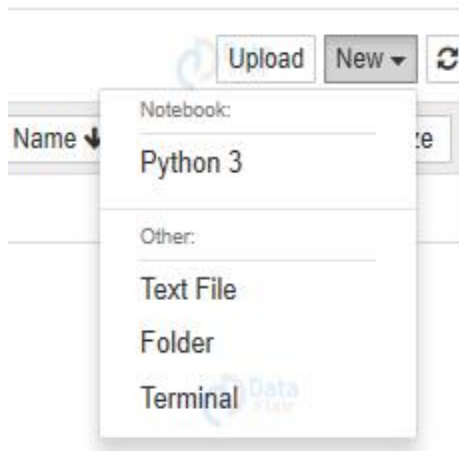
If you go to your command prompt and type 'jupyter notebook', it will open the Jupyter dashboard for you.

```
Administrator: Command Prompt - jupyter notebook

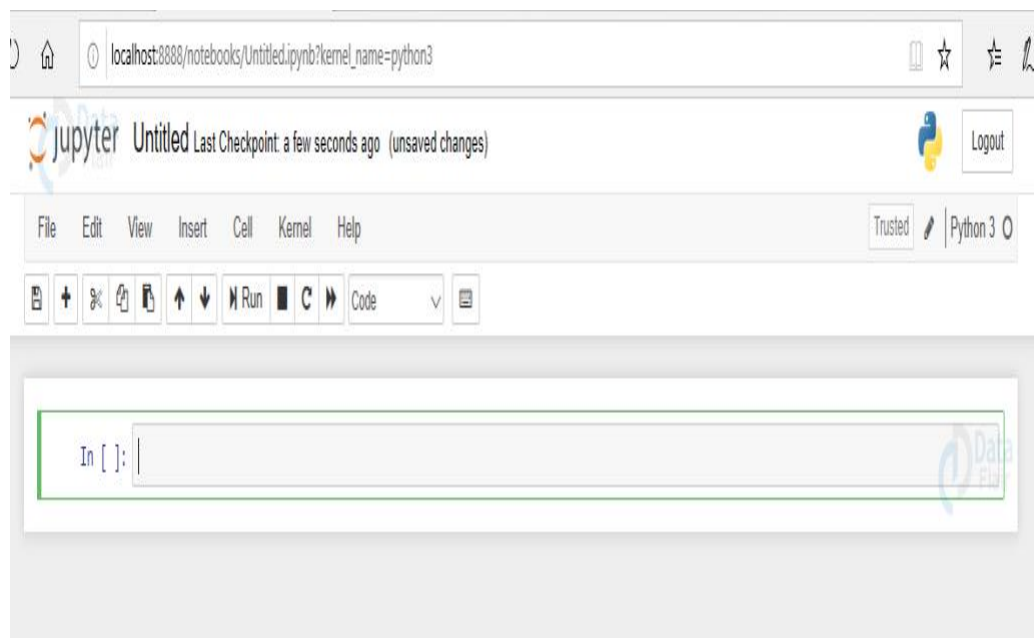
C:\Users\DataFlair>jupyter notebook
[I 11:35:28.378 NotebookApp] JupyterLab extension loaded from c:\users\admin\appdata\local\programs\python\python37\lib\
site-packages\jupyterlab
[I 11:35:28.378 NotebookApp] JupyterLab application directory is c:\users\admin\appdata\local\programs\python\python37\s
hare\jupyterlab
[I 11:35:28.378 NotebookApp] Serving notebooks from local directory: C:\Users\DataFlair
[I 11:35:28.378 NotebookApp] The Jupyter Notebook is running at:
[I 11:35:28.378 NotebookApp] http://localhost:8888/?token=7cffb0fc0d0efb2b1c3358c9cd3cbf63b9fe85328f6551cc
[I 11:35:28.378 NotebookApp] or http://127.0.0.1:8888/?token=7cffb0fc0d0efb2b1c3358c9cd3cbf63b9fe85328f6551cc
[I 11:35:28.378 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:35:28.425 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/ADMIN/AppData/Roaming/jupyter/runtime/nbserver-6308-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=7cffb0fc0d0efb2b1c3358c9cd3cbf63b9fe85328f6551cc
or http://127.0.0.1:8888/?token=7cffb0fc0d0efb2b1c3358c9cd3cbf63b9fe85328f6551cc
```

- You can create a new notebook from the New menu at the top right.



This is your new notebook. You can type in this. To execute a statement or multiple statements at once, press Shift+Enter. Pressing only Enter will only take you to the next line.



## Installing Python Anaconda Libraries

You have successfully installed Anaconda. Now, how will you install libraries or packages with it? Let's see how. Open the Anaconda Prompt and use the 'conda install command' with the package name.



```
Administrator: Anaconda Prompt (Anaconda3) - conda install numpy
(base) C:\Users\ADMIN>conda install numpy
Collecting package metadata (current_repodata.json): done
Solving environment: |
```

```
Administrator: Anaconda Prompt (Anaconda3) - conda install numpy
(base) C:\Users\ADMIN>conda install numpy
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\ProgramData\Anaconda3
  added / updated specs:
    - numpy

The following packages will be downloaded:



| package      | build  |        |
|--------------|--------|--------|
| conda-4.7.12 | py37_0 | 3.0 MB |
| Total:       |        | 3.0 MB |



The following packages will be UPDATED:

  conda                        4.7.10-py37_0 --> 4.7.12-py37_0

Proceed ([y]/n)? _
```

```
Administrator: Anaconda Prompt (Anaconda3)
environment location: C:\ProgramData\Anaconda3
added / updated specs:
  - numpy

The following packages will be downloaded:



| package      | build  |        |
|--------------|--------|--------|
| conda-4.7.12 | py37_0 | 3.0 MB |
| Total:       |        | 3.0 MB |



The following packages will be UPDATED:

  conda                        4.7.10-py37_0 --> 4.7.12-py37_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
conda-4.7.12 | 3.0 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\ADMIN>
```

You must check top python libraries

Now, you can install this package.

```
Administrator: Anaconda Prompt (Anaconda3) - py
- numpy

The following packages will be downloaded:

package | build
-----|-----
conda-4.7.12 | py37_0 3.0 MB
-----|-----
Total: 3.0 MB

The following packages will be UPDATED:

conda 4.7.10-py37_0 --> 4.7.12-py37_0

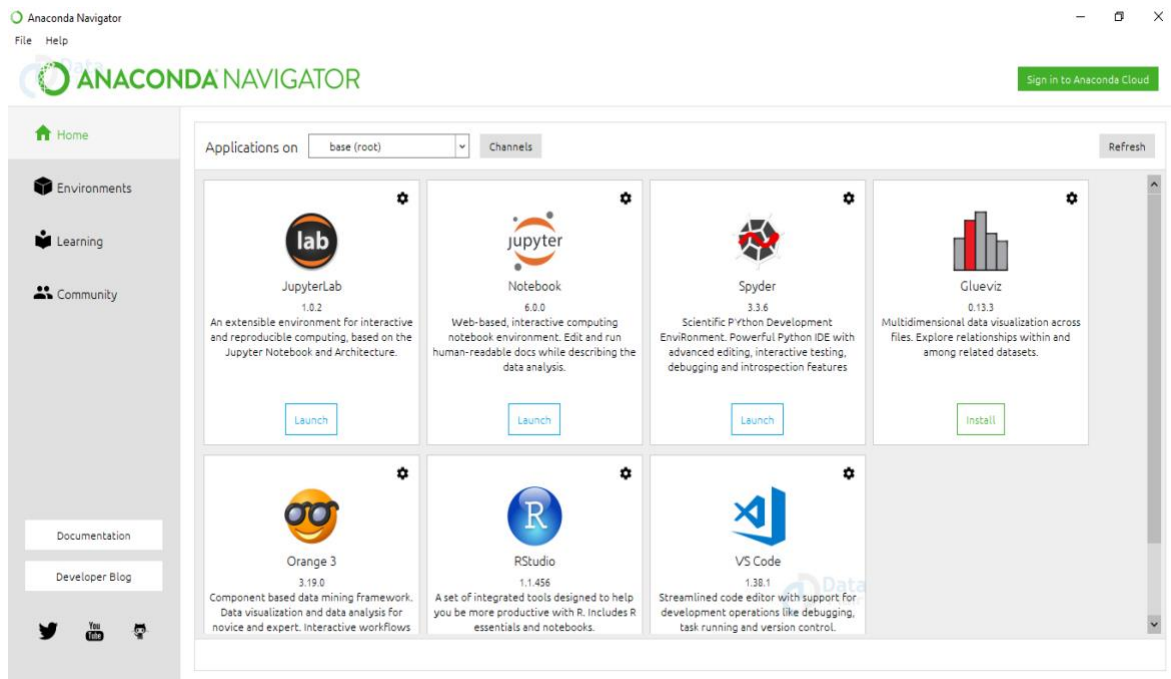
Proceed ([y]/n)? y

Downloading and Extracting Packages
conda-4.7.12 | 3.0 MB | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\ADMIN>py
Python 3.7.4 (tags/v3.7.4:009359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
>>> import numpy
>>>
```

## Anaconda Navigator

Anaconda Navigator is a desktop GUI that ships with Anaconda and lets you launch applications and manage conda packages, environments, and channels without having to use a command-line interface. It can search for packages in a local Anaconda repository or on Anaconda Cloud. With Navigator, you don't need to type commands in a terminal, it lets you work with packages and environments with just a click.



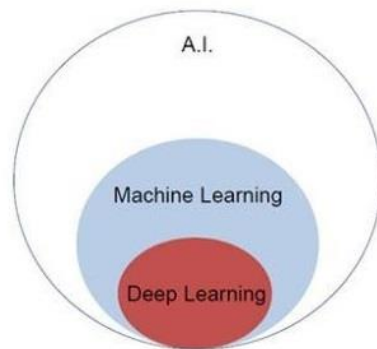
**Fig: 4.8 Anaconda Navigator**

With Navigator, you can access:

- JupyterLab
- Jupyter Notebook
- Spyder

### 4.7.2 Back End Implementation

Artificial Intelligence is one of the most popular trends of recent times. Machine learning and deep learning constitute artificial intelligence. The Venn diagram shown below explains the relationship of machine learning and deep learning.



**Fig: 4.9 Venn diagram**

### Machine Learning

Machine learning is the art of science of getting computers to act as per the algorithms designed and programmed. Many researchers think machine learning is the best way to make progress towards human-level AI. Machine learning includes the following types of patterns.

- Supervised learning pattern
- Unsupervised learning pattern

### Deep Learning

Deep learning is a subfield of machine learning where concerned algorithms are inspired by the structure and function of the brain called artificial neural networks. All the value today of deep learning is through supervised learning or learning from labeled data and algorithms. Each

algorithm in deep learning goes through the same process. It includes a hierarchy of nonlinear transformation of input that can be used to generate a statistical model as output.

Consider the following steps that define the Machine Learning process

- Identifies relevant data sets and prepares them for analysis.
- Chooses the type of algorithm to use.
- Builds an analytical model based on the algorithm used.
- Trains the model on test data sets, revising it as needed.
- Runs the model to generate test scores.

The Two Important Types of Deep Neural Networks

- Convolutional Neural Networks
- Recurrent Neural Networks

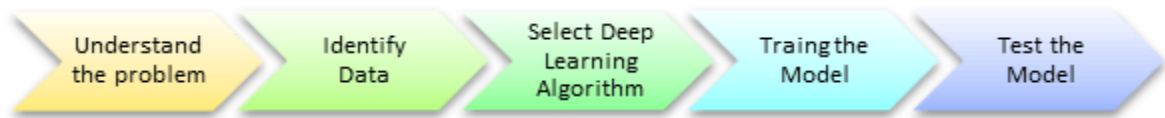
In this we will focus on the CNN, Convolutional Neural Networks.

Deep learning algorithms are constructed with connected layers.

- The first layer is called the Input Layer
- The last layer is called the Output Layer
- All layers in between are called Hidden Layers. The word deep means the network join neurons in more than two layers.
- Each Hidden layer is composed of neurons. The neurons are connected to each other. The neuron will process and then propagate the input signal it receives the layer above it. The strength of the signal given the neuron in the next layer depends on the weight, bias and activation function.
- The network consumes large amounts of input data and operates them through multiple layers; the network can learn increasingly complex features of the data at each layer.

## **Deep learning Process**

A deep neural network provides state-of-the-art accuracy in many tasks, from object detection to speech recognition. They can learn automatically, without predefined knowledge explicitly coded by the programmers.



**Fig: 4.10 Deep Learning Process**

A neural network works quite the same. Each layer represents a deeper level of knowledge, i.e., the hierarchy of knowledge. A neural network with four layers will learn more complex feature than with that with two layers. The learning occurs in two phases.

- Create a statistical model as output.
- The second phase aims at improving the model with a mathematical method known as derivative.

The neural network repeats these two phases hundreds to thousands of time until it has reached a tolerable level of accuracy. The repeat of this two-phase is called an iteration..

## **Convolutional Neural Networks**

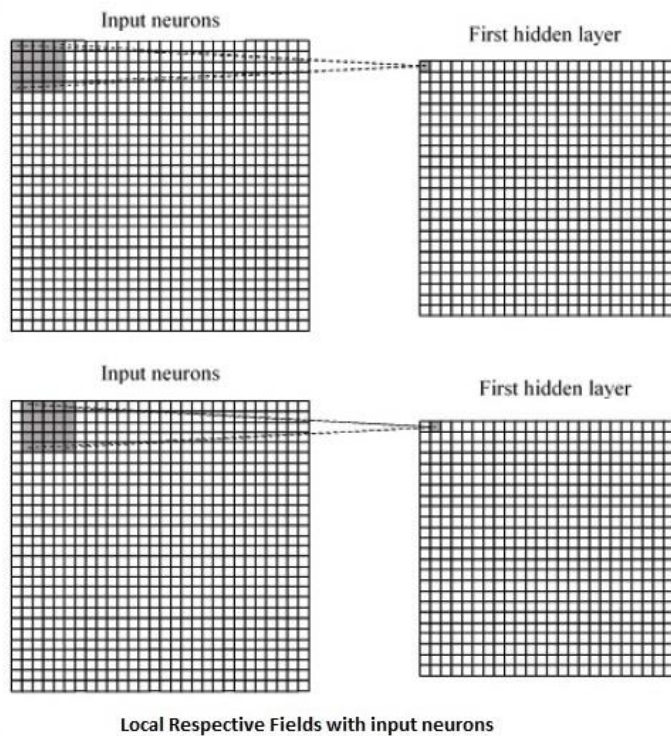
Convolutional Neural networks are designed to process data through multiple layers of arrays. This type of neural networks is used in applications like image Analysis or face recognition. The primary difference between CNN and any other ordinary neural network is that CNN takes input as a two-dimensional array and operates directly on the images rather than focusing on feature extraction which other neural networks focus on.

The dominant approach of CNN includes solutions for problems of recognition. Top companies like Google and Facebook have invested in research and development towards recognition projects to get activities done with greater speed. A convolutional neural network uses three basic ideas :Local respective fields, Convolution, Pooling.

CNN utilizes spatial correlations that exist within the input data. Each concurrent layer of a neural network connects some input neurons. This specific region is called local receptive field. Local receptive field focusses on the hidden neurons. The hidden neurons process the input data inside the mentioned field not realizing the changes outside the specific boundary.

If we observe the above representation, each connection learns a weight of the hidden neuron with an associated connection with movement from one layer to another. Here, individual neurons perform a shift from time to time. This process is called “convolution”.

The mapping of connections from the input layer to the hidden feature map is defined as “shared weights” and bias included is called “shared bias”.



**Fig: 4.11 Local Responsive fields with input neurons**

CNN or convolutional neural networks use pooling layers, which are the layers, positioned immediately after CNN declaration. It takes the input from the user as a feature map that comes out of convolutional networks and prepares a condensed feature map. Pooling layers helps in creating layers with neurons of previous layers.

## Tensor flow

Tensor Flow is a software library or framework, designed by the Google team to implement machine learning and deep learning concepts in the easiest manner. It combines the computational

algebra of optimization techniques for easy calculation of many mathematical expressions. Let us now consider the following important features of Tensor Flow –

- It includes a feature of that defines, optimizes and calculates mathematical expressions easily with the help of multi-dimensional arrays called tensors.
- It includes a programming support of deep neural networks and machine learning techniques.
- It includes a high scalable feature of computation with various data sets.
- Tensor Flow uses GPU computing, automating management. It also includes a unique feature of optimization of same memory and the data used.
- Tensor Flow is well-documented and includes plenty of machine learning libraries. It offers a few important functionalities and methods for the same.
- Tensor Flow is also called a “Google” product. It includes a variety of machine learning and deep learning algorithms. Tensor Flow can train and run deep neural networks for handwritten digit classification, image recognition, word embedding and creation of various sequence models.

## **Tensor Flow – Keras**

One of the most powerful and easy-to-use Python libraries for developing and evaluating deep learning models is Keras; It wraps the efficient numerical computation libraries Theano and Tensor Flow. The advantage of this is mainly that you can get started with neural networks in an easy and fun way. Keras is compact, easy to learn, high-level Python library run on top of TensorFlow framework. It is made with focus of understanding deep learning techniques, such as creating layers for neural networks maintaining the concepts of shapes and mathematical details. The creation of framework can be of the following two types

- Sequential API
- Functional API

Consider the following eight steps to create deep learning model in Keras–

- Loading the data
- Preprocess the loaded data
- Definition of model

- Compiling the model
- Fit the specified model
- Evaluate it
- Make the required predictions
- Save the model

We will use the Jupyter Notebook for execution and display of output.

**Step 1** – Loading the data and preprocessing the loaded data is implemented first to execute the deep learning model.

This step can be defined as “Import libraries and Modules” which means all the libraries and modules are imported as an initial step.

**Step 2** – In this step, we will define the model architecture

**Step 3** – Let us now compile the specified model

**Step 4** – We will now fit the model using training data

The output of iterations created.

## **4.8 Software project Modules**

Implementation is a stage where the theoretical design is automatically converted into programmatically manner. Here we divide the application into number of modules and then coded for deployment. The application is mainly divided into 5 modules.

1. Upload Input Data
2. Data Pre-Processing
3. Segmentation Of Data
4. Training The Data
5. Activity Classification Module



Now let us discuss about each and every module in detail as follows:

#### **4.8.1 Upload Input Data**

In this module we try to construct the application using Python programming language. Once the application is started we try to load input data as one real time video collected from CCTV camera. This video file should be captured from any type of CCTV camera either day mode or night mode and then try to load this input video file.

#### **4.8.2 Data Pre-Processing Module**

Here in this module we try to pre-process the input data by verifying whether it is a valid input data or not. If this is not a valid input the application will not terminate. In this module the input is pre-processed and loaded into the application for next module.

#### **4.8.3 Segmentation of Data Module**

Once the data is pre-processed, now the video file is segmented into frames. This will be dynamically convert the video file into 'N' Number of frames depends on the size of the video data. Once the video file is pre-processed, this will be generating into several frames and all the frames are stored into the buffer location for suspicious activity detection.

#### **4.8.4 Training The Data**

In this module we try to train each and every frame with the several parameters by extracting the front and back ground features present in the frames. Once the front and back ground features are extracted, now the system

can easily identify those frames and then can able to detect the abnormality which is present in that input video.

#### 4.8.5 Activity ClassificationModule

In this module once the data is trained with several features, now the activity classification can be done based on Motion Tracking, head motion tracking, Contact detection and so on. Here we can identify the activity as either Normal activity or Suspicious Activity.

#### 4.9) Sample Code

```
import
cv2

import numpy as np

import math
import itertools

def createMegaBlocks(motionInfoOfFrames,noOfRows,noOfCols):
    n = 2
    megaBlockMotInfVal = np.zeros(((noOfRows/n),(noOfCols/n),len(motionInfoOfFrames),8))
    frameCounter = 0
    for frame in motionInfoOfFrames:
        for index,val in np.ndenumerate(frame[...,:]):
            temp = [list(megaBlockMotInfVal[index[0]/n][index[1]/n][frameCounter]),list(frame[index[0]][index[1]])]
            megaBlockMotInfVal[index[0]/n][index[1]/n][frameCounter] = np.array(map(sum, zip(*temp)))

    frameCounter += 1
    print(((noOfRows/n),(noOfCols/n),len(motionInfoOfFrames)))
    return megaBlockMotInfVal

def kmeans(megaBlockMotInfVal):
    #k-means
    cluster_n = 5
```

```

criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)
flags = cv2.KMEANS_RANDOM_CENTERS
codewords = np.zeros((len(megaBlockMotInfVal),len(megaBlockMotInfVal[0]),cluster_n,8))
#codewords = []
#print("Mega blocks ",megaBlockMotInfVal)
for row in range(len(megaBlockMotInfVal)):
    for col in range(len(megaBlockMotInfVal[row])):
        #print("megaBlockMotInfVal ",(row,col),"/n/n",megaBlockMotInfVal[row][col])
        ret, labels, cw = cv2.kmeans(np.float32(megaBlockMotInfVal[row][col]), cluster_n, None,
criteria,10,flags)
        #print(ret)
        #if(ret == False):
        #    print("K-means failed. Please try again")
        codewords[row][col] = cw
return(codewords)

```

## Motion Detection

```

import
cv2

import numpy as np
import opFlowOfBlocks as roi
import math

def getThresholdDistance(mag,blockSize):
    return mag*blockSize

def getThresholdAngle(ang):
    tAngle = float(math.pi)/2
    return ang+tAngle,ang-tAngle

def getCentreOfBlock(blck1Indx,blck2Indx,centreOfBlocks):
    x1 = centreOfBlocks[blck1Indx[0]][blck1Indx[1]][0]
    y1 = centreOfBlocks[blck1Indx[0]][blck1Indx[1]][1]
    x2 = centreOfBlocks[blck2Indx[0]][blck2Indx[1]][0]
    y2 = centreOfBlocks[blck2Indx[0]][blck2Indx[1]][1]
    slope = float(y2-y1)/(x2-x1) if (x1 != x2) else float("inf")
    return (x1,y1),(x2,y2),slope

```

```

def calcEuclideanDist((x1,y1),(x2,y2)):
    dist = float(((x2-x1)**2 + (y2-y1)**2)**0.5)
    return dist

def angleBtw2Blocks(ang1,ang2):
    if(ang1-ang2 < 0):
        ang1InDeg = math.degrees(ang1)
        ang2InDeg = math.degrees(ang2)
        return math.radians(360 - (ang1InDeg-ang2InDeg))
    return ang1 - ang2

def motionInMapGenerator(opFlowOfBlocks,blockSize,centreOfBlocks,xBlockSize,yBlockSize):
    global frameNo
    motionInfVal = np.zeros((xBlockSize,yBlockSize,8))
    for index,value in np.ndenumerate(opFlowOfBlocks[...]):
        Td = getThresholdDistance(opFlowOfBlocks[index[0]][index[1]][0],blockSize)
        k = opFlowOfBlocks[index[0]][index[1]][1]
        posFi, negFi= getThresholdAngle(math.radians(45*(k)))
        for ind,val in np.ndenumerate(opFlowOfBlocks[...]):
            if(index != ind):
                (x1,y1),(x2,y2), slope = getCentreOfBlock(index,ind,centreOfBlocks)
                euclideanDist = calcEuclideanDist((x1,y1),(x2,y2))
                if(euclideanDist< Td):
                    angWithXAxis = math.atan(slope)
                    angBtwTwoBlocks = angleBtw2Blocks(math.radians(45*(k)),angWithXAxis)
                    if(negFi<angBtwTwoBlocks and angBtwTwoBlocks<posFi):
                        motionInfVal[ind[0]][ind[1]][int(opFlowOfBlocks[index[0]][index[1]][1])] += math.exp(-
1*(float(euclideanDist)/opFlowOfBlocks[index[0]][index[1]][0]))
                    #print("Frame number ", frameNo)
            frameNo += 1
    return motionInfVal

def getMotionInfluenceMap(vid):
    global frameNo
    frameNo = 0
    cap = cv2.VideoCapture(vid)
    ret, frame1 = cap.read()
    rows, cols = frame1.shape[0], frame1.shape[1]
    print(rows,cols)
    prvs = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
    motionInfOfFrames = []
    count = 0
    while 1:

```

```

'''
    #if(count <= 475 or (count > 623 and count <= 1300)):
if(count < 475):
    ret, frame2 = cap.read()
    prvs = cv2.cvtColor(frame2,cv2.COLOR_BGR2GRAY)
    count += 1
    continue
'''

    #if((count < 1451 and count <= 623)):
'''

if(count < 475):
    ret, frame2 = cap.read()
    prvs = cv2.cvtColor(frame2,cv2.COLOR_BGR2GRAY)
    count += 1
    continue
'''

    print(count)
    ret, frame2 = cap.read()
    if (ret == False):
        break
    next = cv2.cvtColor(frame2, cv2.COLOR_BGR2GRAY)
    flow = cv2.calcOpticalFlowFarneback(prvs, next, None, 0.5, 3, 15, 3, 5, 1.2, 0)
    mag, ang = cv2.cartToPolar(flow[...], flow[...])
    prvs = next
    opFlowOfBlocks,noOfRowInBlock,noOfColInBlock,blockSize,centreOfBlocks,xBlockSize,yBlockSize =
    roi.calcOptFlowOfBlocks(mag,ang,next)
    motionInfVal = motionInMapGenerator(opFlowOfBlocks,blockSize,centreOfBlocks,xBlockSize,yBlockSize)
    motionInfOfFrames.append(motionInfVal)
    #if(count == 622):
    #    break
    count += 1
    return motionInfOfFrames, xBlockSize,yBlockSize
'''

```

## Testing.py File

```

import
motionInfluenceGenerator as
mig

import createMegaBlocks as cmb
import numpy as np
import cv2

```

```

def square(a):
    return (a**2)

def diff(l):
    return (l[0] - l[1])

def showUnusualActivities(unusual, vid, noOfRows, noOfCols, n):
    unusualFrames = unusual.keys()
    unusualFrames.sort()
    print(unusualFrames)
    cap = cv2.VideoCapture(vid)
    ret, frame = cap.read()
    rows, cols = frame.shape[0], frame.shape[1]
    rowLength = rows/(noOfRows/n)
    colLength = cols/(noOfCols/n)
    print("Block Size ",(rowLength,colLength))
    count = 0
    screen_res = 980, 520
    scale_width = screen_res[0] / 320
    scale_height = screen_res[1] / 240
    scale = min(scale_width, scale_height)
    window_width = int(320 * scale)
    window_height = int(240 * scale)

    cv2.namedWindow('Unusual Frame',cv2.WINDOW_NORMAL)
    cv2.resizeWindow('Unusual Frame',window_width, window_height)
    while 1:
        print(count)
        ret, uFrame = cap.read()
        ""

    if(count <= 475):
        count += 1
        continue
    elif((count-475) in unusualFrames):
        ""

    if(count in unusualFrames):
        if (ret == False):
            break
        for blockNum in unusual[count]:
            print(blockNum)
            x1 = blockNum[1] * rowLength
            y1 = blockNum[0] * colLength
            x2 = (blockNum[1]+1) * rowLength
            y2 = (blockNum[0]+1) * colLength

```

```

        cv2.rectangle(uFrame,(x1,y1),(x2,y2),(0,0,255),1)
print("Unusual frame number ",str(count))
    cv2.imshow('Unusual Frame',uFrame)
    cv2.waitKey(0)
    #cv2.destroyAllWindows()
'''
if(count == 622):
    break
'''
    count += 1
def constructMinDistMatrix(megaBlockMotInfVal,codewords, noOfRows, noOfCols,
vid):
    #threshold = 2.1874939946e-21
    #threshold = 0.00196777849633
    #threshold = 9.3985643749758953e-06
    #threshold = 0.439167467697
    #threshold = 0.021305195096797892
    #threshold = 3.35845489394e-07
    #threshold = 1.6586380629e-08
    #threshold = 0.000212282134156
    #threshold = 4.63266766923e-14
    #threshold = 7.29868038369e-06
    #threshold = 8.82926005091e-05
    #threshold = 7.39718222289e-14
    #threshold = 8.82926005091e-05
    #threshold = 0.0080168593265873295
    #threshold = 0.00511863986892
    #-----#
    threshold = 5.83682407063e-05
    #threshold = 3.37029584538e-07
    #-----#
    #threshold = 2.63426664698e-06
    #threshold = 1.91130257263e-08
    #threshold = 0.0012675861679
    #threshold = 1.01827939172e-05
    n = 2
    minDistMatrix =
    np.zeros((len(megaBlockMotInfVal[0][0]),(noOfRows/n),(noOfCols/n)))
    for index,val in np.ndenumerate(megaBlockMotInfVal[...0]):
    euclidianDist = []
        for codeword in codewords[index[0]][index[1]]:
            #print("haha")
            temp = [list(megaBlockMotInfVal[index[0]][index[1]][index[2]]),list(codeword)]

```

```

        #print("Temp",temp)
dist = np.linalg.norm(megaBlockMotInfVal[index[0]][index[1]][index[2]]-codeword)
        #print("Dist ",dist)
eucDist = (sum(map(square,map(diff,zip(*temp)))))**0.5
        #eucDist = (sum(map(square,map(diff,zip(*temp)))))
eucledianDist.append(eucDist)
        #print("My calc ",sum(map(square,map(diff,zip(*temp)))))
        #print(min(eucledianDist))
minDistMatrix[index[2]][index[0]][index[1]] = min(eucledianDist)
        unusual = { }
        for i in range(len(minDistMatrix)):
            if(np.amax(minDistMatrix[i]) > threshold):
                unusual[i] = []
                for index,val in np.ndenumerate(minDistMatrix[i]):
                    #print("MotInfVal_train",val)
if(val> threshold):
                    unusual[i].append((index[0],index[1]))
                print(unusual)
showUnusualActivities(unusual, vid, noOfRows, noOfCols, n)
def test_video(vid):
    """
        calls all methods to test the given video
    """
    print "Test video ", vid
MotionInfOfFrames, rows, cols = mig.getMotionInfluenceMap(vid)
    #np.save("videos\scene1\rows_cols_set1_p1_test_20-20_k5.npy",np.array([rows,cols]))
    #####print "Motion Inf Map ", len(MotionInfOfFrames)
    #numpy.save("MotionInfluenceMaps", np.array(MotionInfOfFrames),
allow_pickle=True, fix_imports=True)
megaBlockMotInfVal = cmb.createMegaBlocks(MotionInfOfFrames, rows, cols)
    #####rows, cols = np.load("rows_cols__set3_p2_test_40_k3.npy")
    #print(megaBlockMotInfVal)
np.save("videos\scene1\megaBlockMotInfVal_set1_p1_test_20-
20_k5.npy",megaBlockMotInfVal)
    #####megaBlockMotInfVal =
np.load("megaBlockMotInfVal_set3_p2_train_40_k7.npy")
    codewords = np.load("videos\scene1\codewords_set2_p1_train_20-20_k5.npy")
    print("codewords",codewords)
listOfUnusualFrames = constructMinDistMatrix(megaBlockMotInfVal,codewords,rows,
cols, vid)
    return
if __name__ == '__main__':
    """

```



```
        defines training set and calls trainFromVideo for every vid
    ""
    for video in testSet:
test_video(video)
    print "Done"
```

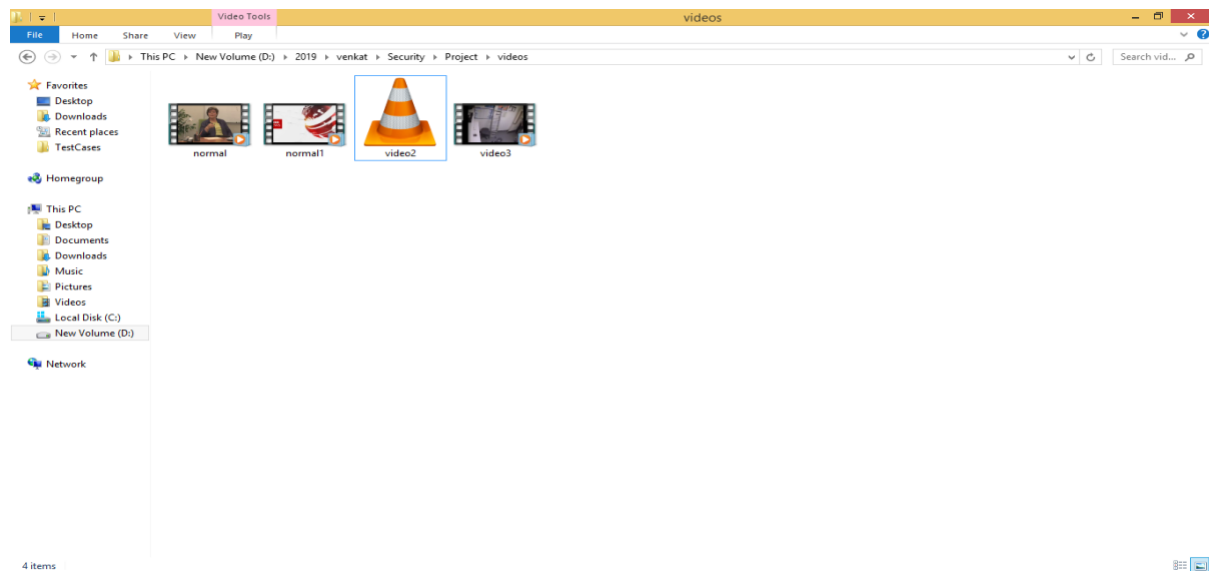
# **CHAPTER 5**

## **EXPERIMENTAL RESULTS**

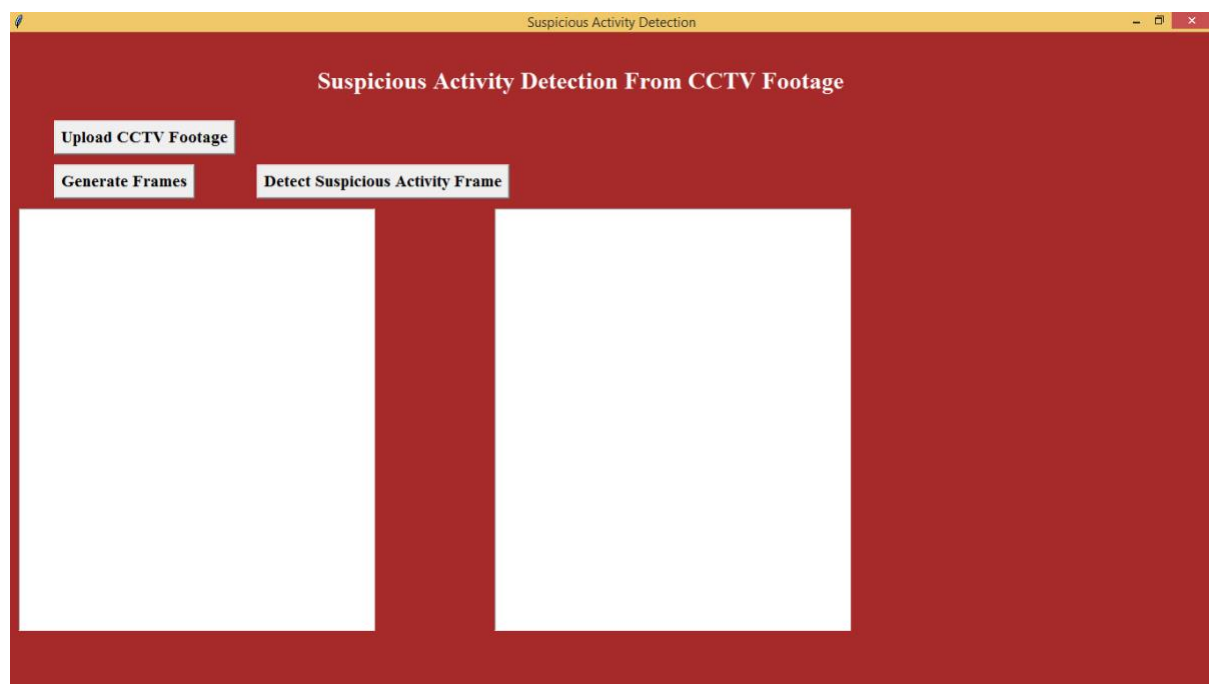
### **5.1 RESULTS (OUTPUT SCREENS)**

#### **OUTPUT SCREENS**

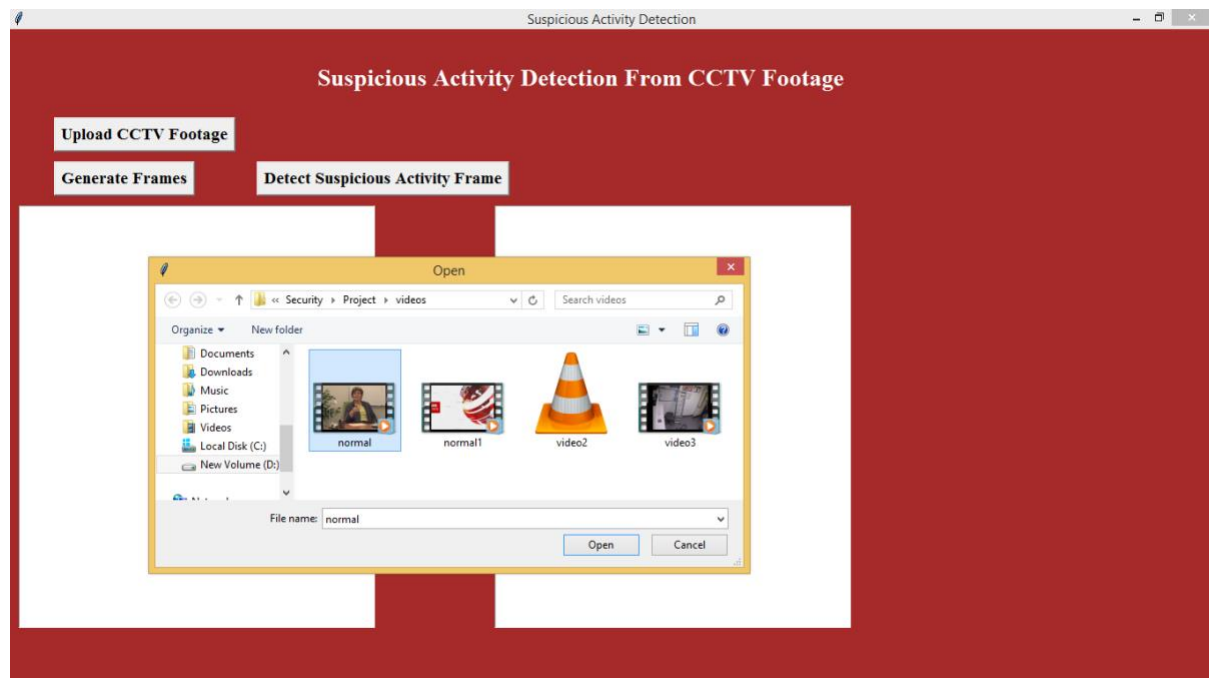
After executing above commands all software will be installed. To monitor i am using below videos. For training i used human images who cover their faces to perform suspicious activity and if any video contains person covering their faces then application will detect it as a suspicious activity.



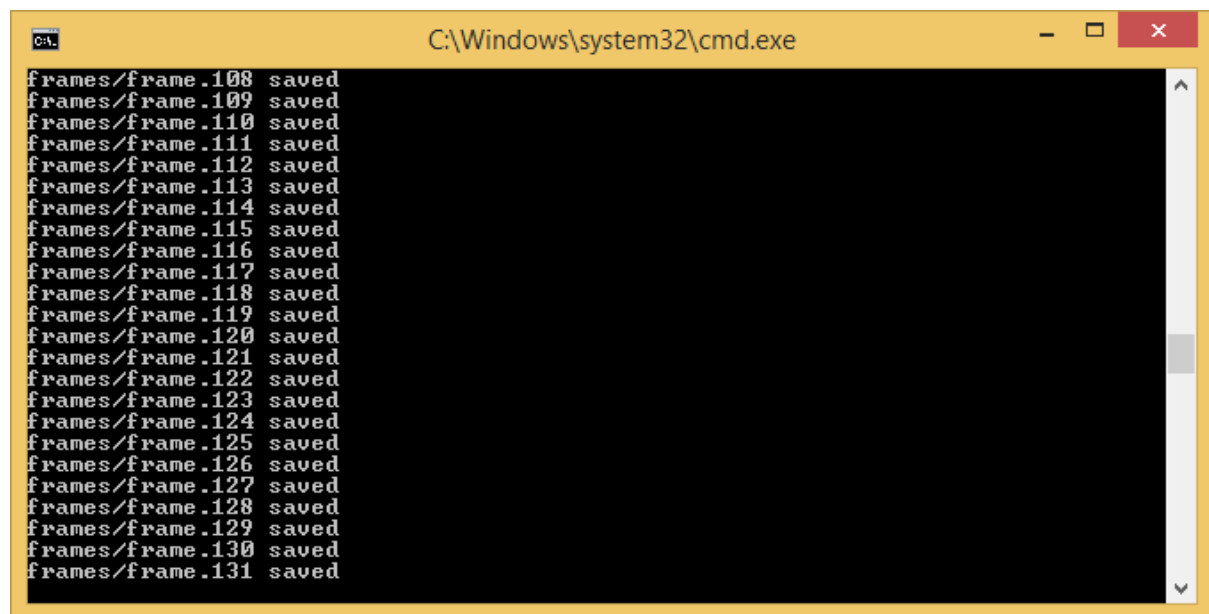
Double click on ‘run.bat’ file from project folder to start project execution. We will get below screen



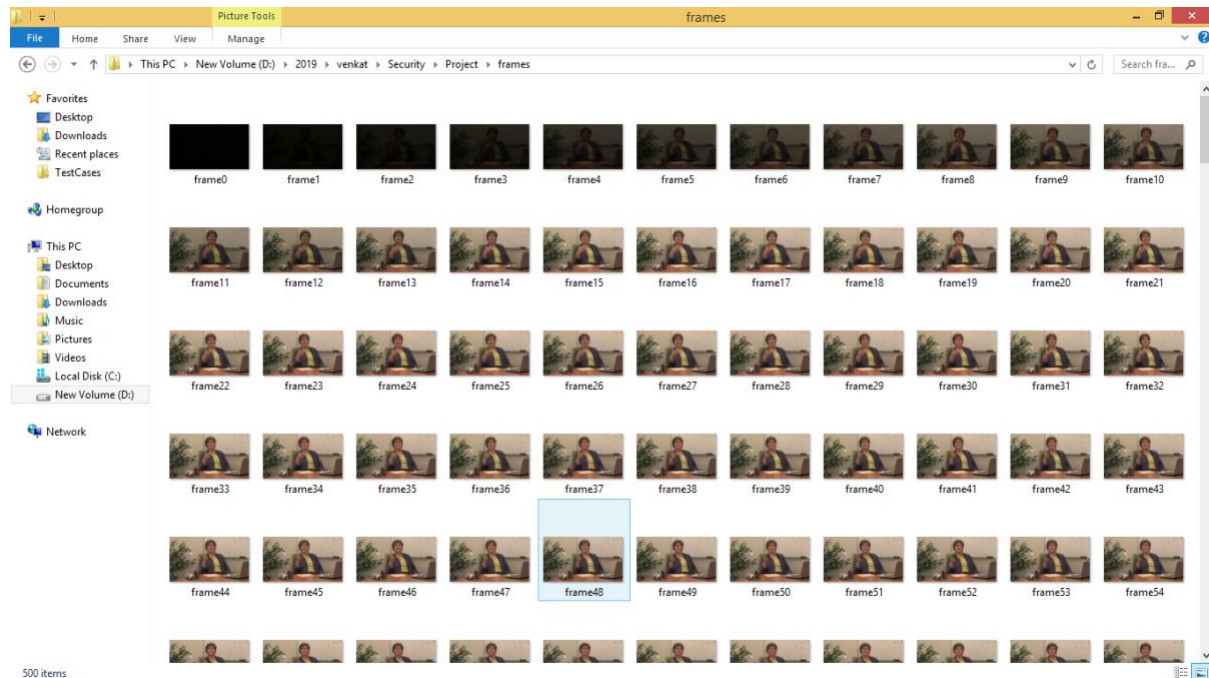
Click on ‘Upload CCTV Footage’ button to upload video



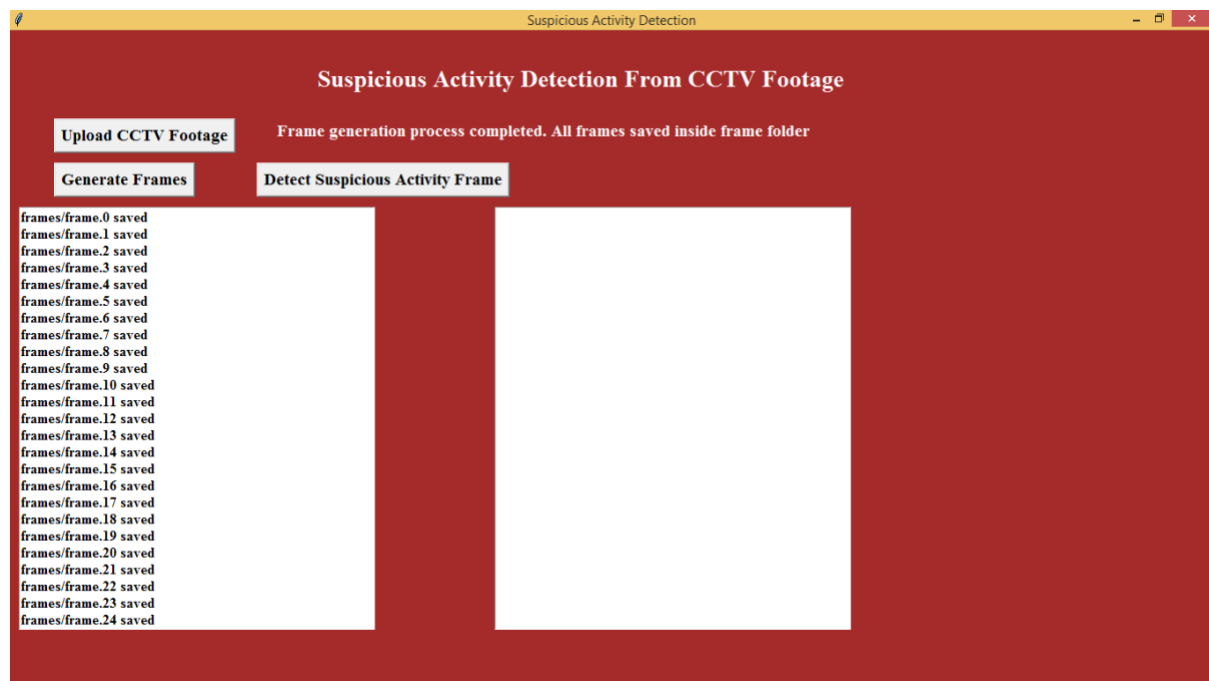
In above screen i am uploading one normal video. After uploading video click on 'Generate Frames' button to generate frame



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

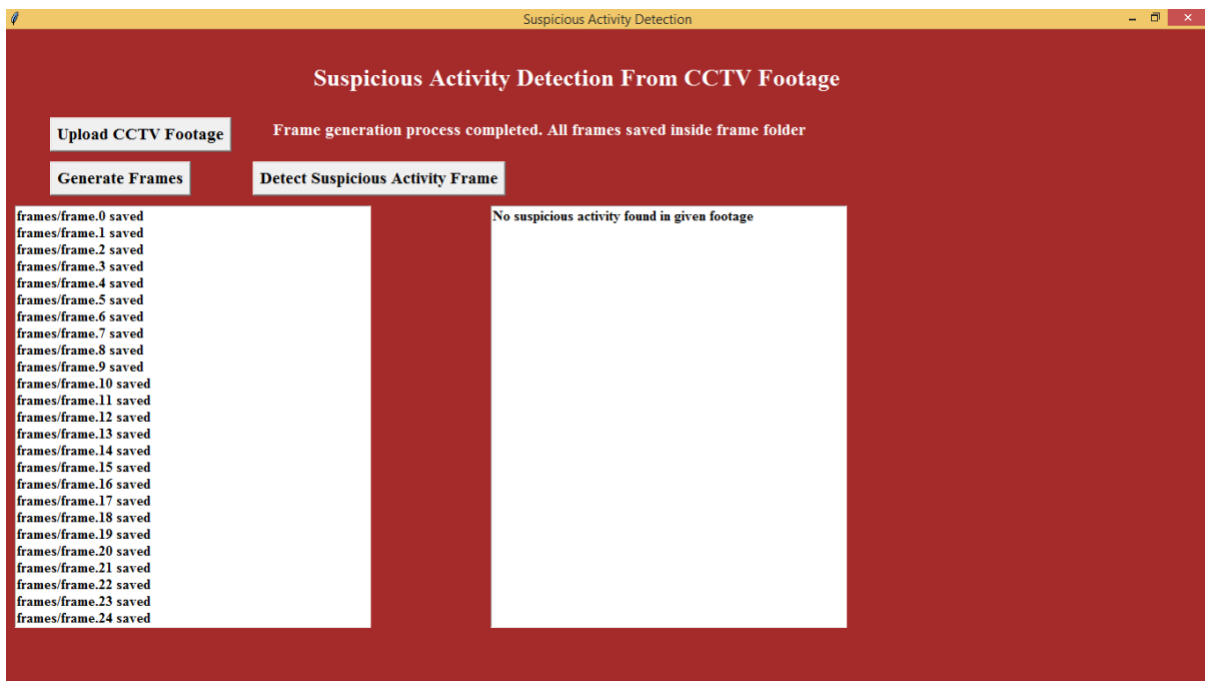


Now click on 'Detect Suspicious Activity Frame' button to start monitoring frames for suspicious activity

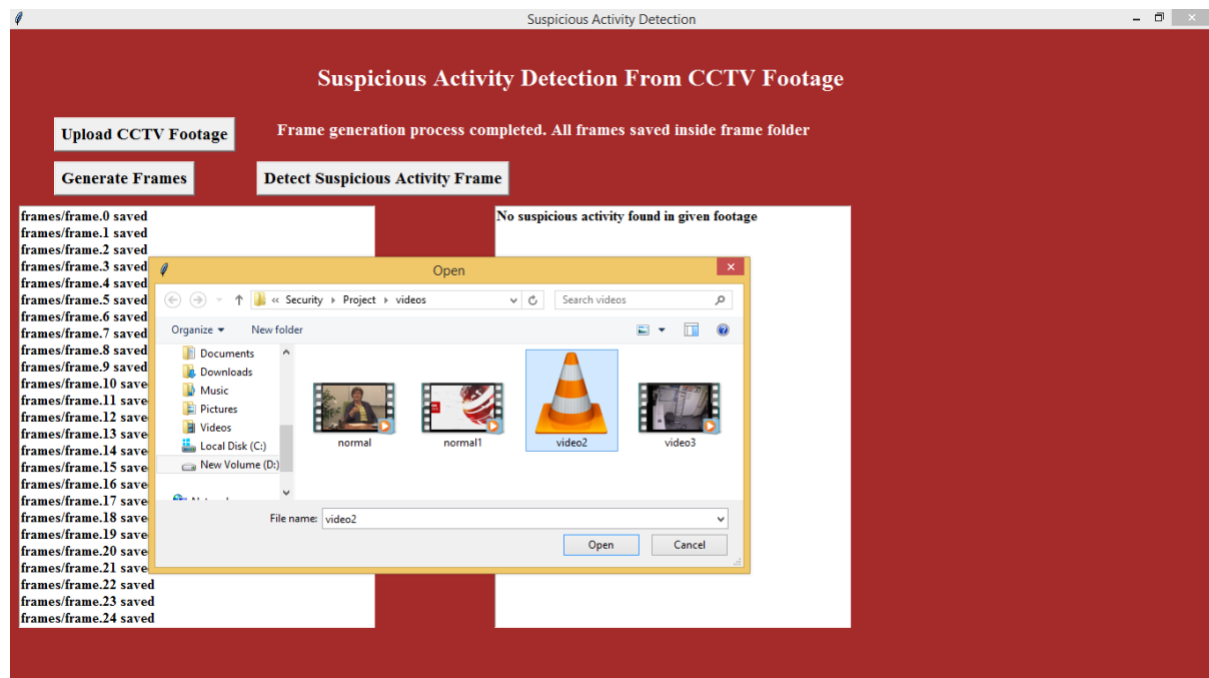
```
C:\Windows\system32\cmd.exe

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
```

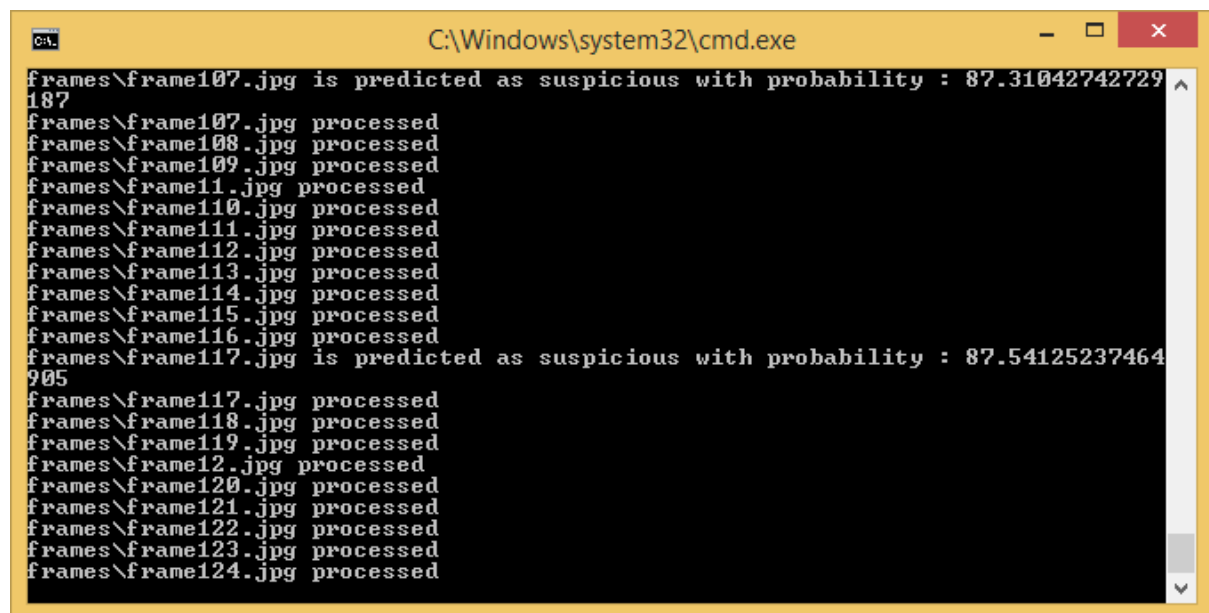
In above black console window we can see processing of each frame to detect suspicious activity.



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



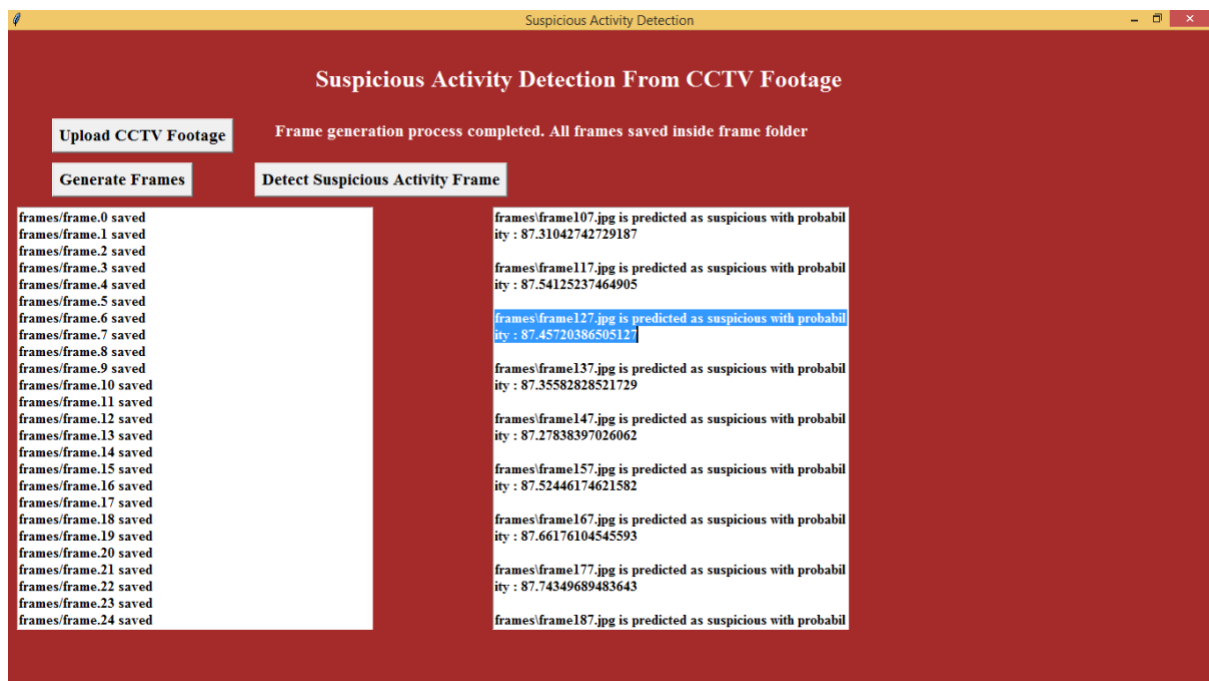
In above screen i am uploading 'Video2' and then extract frames



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 which has such activities.

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



## **5.2 TESTING AND TEST CASES**

### **5.2 Testing**

Verification and Validation encompass a wide range of software quality assurance(SQA) activities that include formal, technical review and quality and configuration audits, performance monitoring,simulation,feasibility study,documentation,review algorithm analysis, development testing, qualification testing and installation testing.

#### **5.2.1 Verification**

Verification is the fundamental concept in the software design. Verification refer in the set of activities that ensure the software correctly implement a specification function validation refer to a different set of activities that ensure that software that has been built in traceable customer. Validation is nothing but the process of using software in live environment in order to find errors verification is typically done in two steps.

- Verification of the requirement verification that the software requirement definition satisfies the customer needs.
- Verification of the design: verification that the design satisfies the requirement definition.

Software reviews are one of the most important SQA activities. Review serves as a filter for the software process, removing errors while they are relatively inexpensive to find and correct. To properly verify a system, data about engineering process should be collect, eventually and disseminated,SQA helps to improve the quality of the product and the software procedure.

#### **5.2.2 Testing Methodology**

Testing a is process, which reveals in the program. It is the condition known as text cased and the e major quality measure employed during software development. During testing, the program is executed with set of condition known as text cased and output evaluated to determine whether the program is performing as expected. In order to make sure that the system does not error, the different levels of testing strategies that are applied at different levels of testing strategies that are applied at different phases of software development are:

### **5.2.1.1 UNIT TESTING**

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer requirement.

### **5.2.1.2 BLACK BOX TESTING**

In this strategy some test cases are generated as input condition that execute all function requirement for the program this testing has been used to find error in the following categories:

- Incorrect or missing function
- Interface errors
- Error in data structure or external database access
- Performance
- Initialization

In this testing the output is checked for correctness the logical flow of the data is not checked

### **5.2.1.3 WHITE BOX TESTING**

In this test cases are generated on the logic of each module by drawing flow graph of that module and logical decision are tested on all the cases. While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test.

- Guarantee that all independent paths have been executed.
- Executed all logic decision on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structures to ensure their validity.

### **5.2.1.4 INTEGRATION TESTING**

Integration testing ensures that software and subsystem work together as whole. It tests the interface of the entire module to make behave properly when integrated together.

#### **5.2.1.5 SYSTEM TESTING**

Involves in-housing testing of entire system before delivery to the user. It aim is to satisfy the user the system meet all requirement of the clients specification.

#### **5.2.1.6 VALIDATION TESTING**

The system has been tested and implemented successfully and thus ensured that all requirement are listed in the software requirement specification are completely fullfilled.In case of enormous input corresponding error message are displayed

#### **5.2.1.7 COMPILATION TESTING**

It was a good idea to do our stress early on, because it gave us time to fix some unexpected deadlocks and stability problem that only occurred when components were exposed to very high transaction

Volumes.

#### **5.2.1.8 EXECUTIVE TESTING**

This program was successfully loaded and executed, because of good programming there no execution errors.

#### **5.2.1.9 OUTPUT TESTING**

The successful output screens are placed in the output screens section above.

#### **5.2.1.10 ACTIVITIES UNDERTAKEN DURING INTEGRATION AND SYSTEM TESTING**

It involves testing of integrating various modules and testing their overall performance due to their integration. Integration of different modules is undertaken once they have been coded and unit tested. During the integration and system testing phase, the modules are integrated in a planned manner and the goal of system testing is to ensure that the developed system conforms to its requirements laid out in the SRS document. System testing usually consists of three different kinds of testing activities:

- $\alpha$  – testing: It is the system testing performed by the development team.
- $\beta$  – Testing: It is the system testing performed by a friendly set of customers.
- acceptance testing: It is the system testing performed by the customer himself after the

### **5.3 Test Cases for Unit Testing and Validation Testing's**

S.No	TCID	DESCRIPTION/ SAMPLE INPUT	EXPECTED VALUE	OBSERVED VALUE	Result
1	Tc1	User try to load valid data	If the user choose CCTV footage video	Display Message as "Data Loaded Successfully"	Success
2	Tc2	User Failed to load valid data	If the user failed to choose CCTV footage video data	Display Error Message as " Data Loading Failed"	Success
3	Tc3	User Generate Frames	Once data is loaded, user will generate frames	Display Message as "Frames are Generated"	Success
4	Tc4	User Fail to Generate Frames	If the user don't choose valid data, frames are not generated	Display Error Message as " Frames are not Generated"	Success
5	TC5	Activity Detection	User try to detect activity	Display Message as "Activity Detected"	Success

# **CHAPTER 6**

## **CONCLUSION AND FUTURE SCOPE**

## **6. CONCLUSION AND FUTURE WORK**

In this proposed work, we have discussed the various techniques related to abandoned object detection, theft detection, falling detection, accidents and illegal parking detection, violence detection and fire detection for the foreground object extraction, tracking, feature extraction and classification. In past decades, several researchers proposed novel approaches with noise removal, illumination handling, and occlusion handling methods to reduce the false object detection. Many researchers have also worked for making real-time intelligent surveillance system but processing rate of the video frames is not as good as required and there is no such system that has been developed with 100% detection accuracy and 0% false detection rate for videos having complex background.

## 7. REFERENCES

1. N. Ihaddadene and C. Djeraba., (2008), “Real-time crowd motion analysis”, IEEE, pp 1-4.
2. K. Yokoi, T. Watanabe, and S. Ito (2009), “Surveillance Event Detection Task”, pp 1-10.
3. W. Hu, T. Tan, L. Wang, and S. Maybank (2004), “A Survey on Visual Surveillance of Object Motion and Behaviours”. IEEE, Vol: 34, Issue 3, pp 334 – 352.
4. Divya J, M.E, G. H, Pune, Prof. Dr. R.S.Bichkar (2015),” Automatic Video Based Surveillance System for Abnormal Behaviour Detection”. IJSR, Vol: 4 Issue: 7, pp 1743- 1747.
5. Kaushik Deb, Sayem Imtiaz, and Priyam Biswas (2014),” A Motion Region Detection and Tracking Method”, vol. 4, no. 1, pp79-90.
6. Prof. D. D. Dighe, Ms. K. V. Patil (2014), “Background Subtraction Algorithm Based Human Behaviour Detection “,Vol. 4, Issue 7( Version 3), pp.60-64,ijera.
7. Srinivasa Rao Chalamala, Nisha Pal, Suneeta Agarwal (2015), “Detection and Tracking of Moving Objects in Surveillance System”.Volume.14, pp 2321- 1814, ISBN (Print): 978-1-62951-946-3. 84
8. R.Naveen Kumar and Dr.S.Chandrakala (2016),”DETECTING AGGRESSIVE HUMAN BEHAVIOR IN PUBLIC ENVIRONMENTS “Department of computer science and engineer, Rajalakshmi Engineering College Chennai, Tamil nadu. ISSN: 0976-1353 Volume 22 Issue 2.
9. Pooja N S., (2015), “Suspicious Group Event Detection for Outdoor Environment”, (IJMTER), Volume 0X, Issue 0Y, pp 611-618.
10. C.Srinivas Rao, P.Darwin (2012), “Frame Difference and Kalman Filter Techniques for Detection of Moving Vehicles in Video Surveillance”, Vol. 2, Issue 6, pp.1168-1170, (IJERA).
11. D STALIN ALEX, Dr. AMITABH WAHI, (2014),”BSFD: BACKGROUND SUBTRACTION FRAME DIFFERENCE ALGORITHM FOR MOVING OBJECT DETECTION AND EXTRACTION”, Vol. 60 No.3, JATIT & LLS.



12. Christopher M. Bishop, M. Jordan, J.Kleinberg, B. Schölkopf (2006), “Pattern Recognition and Machine Learning” Springer.
13. Yung-Tai Hsu, Hong-Yuan Mark Liao, Jun-Wei Hsieh, Chih-Chiang Chen. (2008), “Video-Based Human Movement Analysis and Its Application to Surveillance Systems”, IEEE, VOL. 10, Issue: 3, pp.372-384.
14. Chitra Hegde, Shakti Singh Chundawat, Divya S N (2016), “Unusual Event Detection Using Mean Feature Point Matching Algorithm”.Vol.6, pp.1595- 1601,ISSN: 2088-8708, DOI: 10.11591/ijece.v6i4.10179.
15. R.T. Collins and et al (2000), “A system for video surveillance and monitoring”. Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.