**1. INTRODUCTION**

Coronavirus is a large family virus that harms humans and animals. Covid-19 is known as a family member of coronavirus, first spread to Wuhan, China in December 2019. The outbreak then rapidly affected many countries in the world and had been declared as a pandemic by the World Health organization (WHO).

Based on the information from WHO, the coronavirus is spreading from a person to a person via small droplets from the nose and mouth. In other words, social distancing is the best practice where people can minimize physical contact with possible coronavirus carriers, by keeping the distance at least one meter away from each other.

This project is proposed to support the actions on Covid19 spread mitigation. It provides a solution for detecting people gathering in public places such as banks, shopping malls, clinics etc. The concept of person detection algorithm is used to accurately detect a person’s presence in areas of interest and is then followed by measuring the distance between the detected persons.

**1.1 Objectives**

The objective is to reduce transmission, delaying the epidemic peak, reducing the size of the epidemic peak, and spreading cases over a longer time to relieve pressure on the healthcare system. It is an action taken to minimize contact with other individuals. It has been suggested that maintaining a distance of approximately 2 metres from another individual result in a marked reduction in transmission of most flu virus strains, including COVID-19.

**1.2 Methodology**

**The Proposed System**

The proposed system focuses on how to identify the person on image/video stream whether the social distancing is maintained or not with the help of yolo and deep learning algorithm by using the OpenCV and few other libraries.

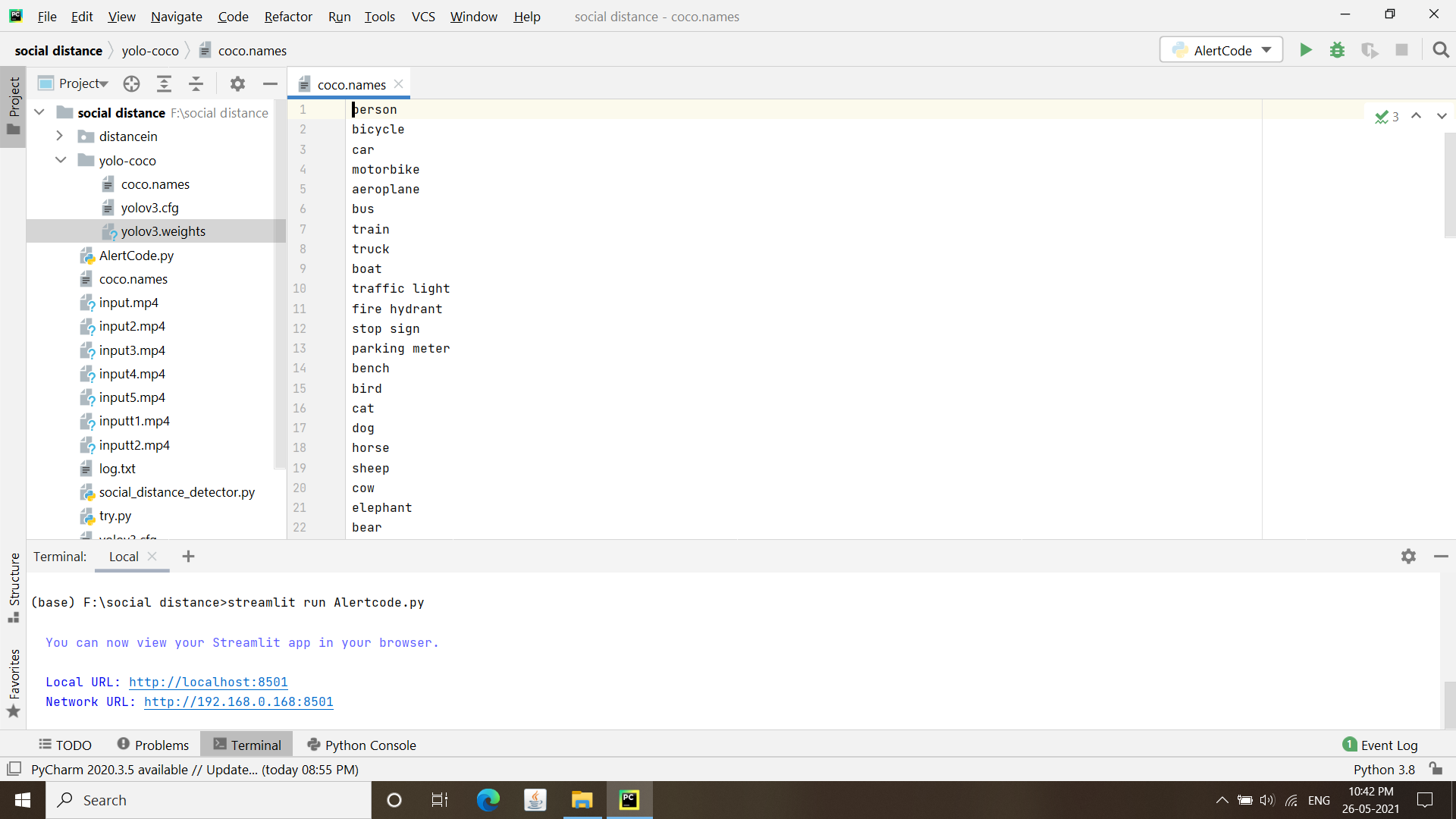
**1.2.1 Dataset**

The MS **COCO** (**Microsoft Common Objects in Context**) dataset is a large-scale object detection, segmentation, key-point detection, and captioning dataset. The dataset consists of 328K images. As hinted by the name, images in COCO dataset are taken from everyday scenes thus attaching “context” to the objects captured in the scenes. We can put an analogy to explain this further. Let’s say we want to detect a person object in an image. A non-contextual, isolated image will be a close-up photograph of a person. Looking at the photograph, we can only tell that it is an image of a person. However, it will be challenging to describe the environment where the photograph was taken without having other supplementary images that capture not only the person but also the studio or surrounding scene.

**Annotations:** The dataset has annotations for

* object detection: bounding boxes and per-instance segmentation masks with 80 object categories,
* captioning: natural language descriptions of the images (see MS COCO Captions),

keypoints detection: containing more than 200,000 images and 250,000 person instances labeled with keypoints (17 possible keypoints, such as left eye, nose,right hip, right ankle)

****

**Fig: 1.2.1 Some of the labels in COCO Dataset**

**1.2.2 YOLO model**

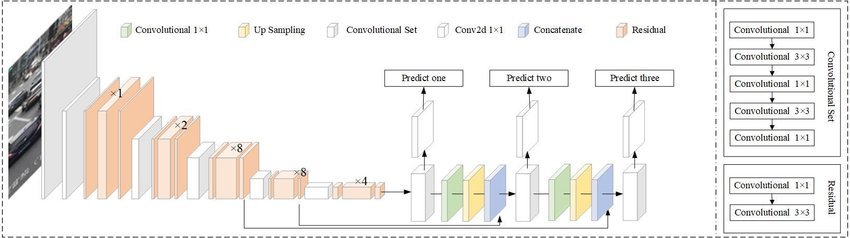
YOLO stands for You Only Look Once and is quite possibly the most famous item indicator convolutional neural organizations (CNNs). It is a pre-trained model, which is trained on COCO dataset. It is a real-time object detection algorithm that identifies specific objects in videos, live feeds, or images. Versions 1-3 of YOLO were created by Joseph Redmon and Ali Farhadi. The first version of YOLO was created in 2016, and version 3, which is discussed extensively in this article, was made two years later in 2018. YOLO is implemented using the Keras or OpenCV deep learning libraries.

YOLO is based on the darknet, built in C. [Darknet](https://pjreddie.com/darknet/) is an open source neural network framework written in C and CUDA.

**Working of YOLO**

YOLO is a fully Convolutional Neural Network(CNN) used for object detection. CNN’s are classifier-based systems that can process input images as structured arrays of data and identify patterns between them. YOLO has the advantage of being much faster than other networks and still maintains accuracy.

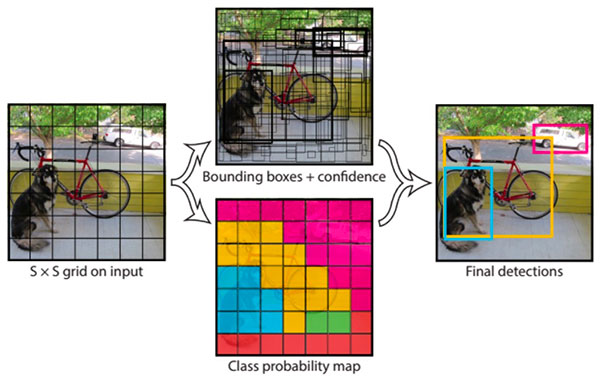
It allows the model to look at the whole image at test time so its predictions are informed by the global context in the image. YOLO and other convolutional neural network algorithms “score” regions based on their similarities to predefined classes. High-scoring regions are noted as positive detections of whatever class they most closely identify with. For example, in a live feed of traffic, YOLO can be used to detect different kinds of vehicles depending on which regions of the video score highly in comparison to predefined classes of vehicles.



**Fig 1.2.2.1: YOLO Model**

**YOLO Detection**

The YOLOv3 algorithm first separates an image into a grid. Each grid cell predicts some number of boundary boxes (sometimes referred to as anchor boxes) around objects that score highly with the aforementioned predefined classes. Each boundary box has a respective confidence score of how accurate it assumes that prediction should be, and detects only one object per bounding box. The boundary boxes are generated by clustering the dimensions of the ground truth boxes from the original dataset to find the most common shapes and sizes.



**Fig 1.2.2.2: YOLO Detection**

**Advantages of YOLO Model**

* Process frames at the rate of 45fps to 150fps which is better than real-time.
* The network is able to generalize image better.

**Disadvantages** **of** **YOLO** **Model**

* Struggles to detect close objects because each grid can propose only 2 bounding boxes.
* Struggles to detect small objects.

**1.2.3 Deep Neural Network**

A deep neural network (DNN) is an ANN with multiple hidden layers between the input and output layers. The main purpose of a neural network is to receive a set of inputs, perform progressively complex calculations on them, and give output to solve real world problems like classification.

A DNN model is a feed forward neural network with multiple layers which are nonlinear transformers. It transforms input vectors to corresponding output vectors and facilitates subsequent recognition. For an Automatic Speech Recognition (ASR) system, a DNN model is used to transform features ofspeeches to labels corresponding to states in Hidden Markov Models (HMMs) .

Deep neural networks are a powerful category of machine learning algorithms implemented by stacking layers of neural networks along the depth and width of smaller architectures. Deep neural networks with several layers have recently become a highly successful and popular research topic in machine learning due to their excellent performance in many benchmark problems and applications.

**Architecture**

The structure of a DNN model generally consists of the following parts :

(i) An input layer which restores the inputs from the features;

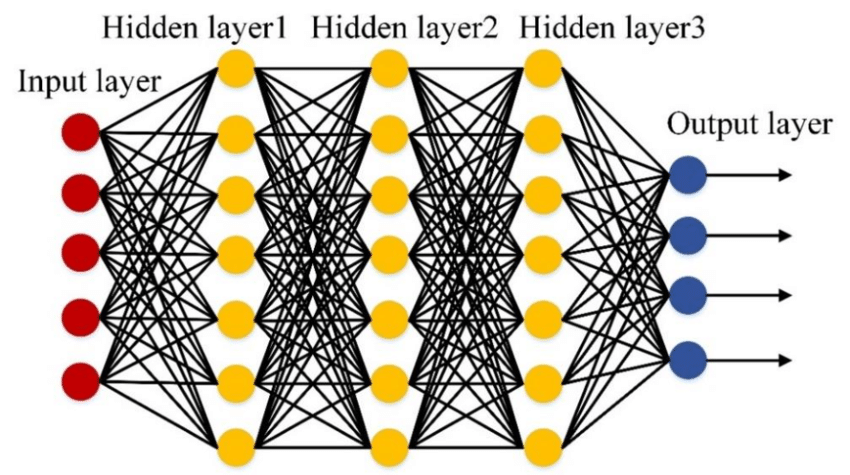
(ii) Hidden units which restore the intermediate transforming results;

(iii) An output layer which restores the outputs representing the final transforming results;

(iv) Activation functions which simulate properties of biological neurons;

(v) Weights which represent the importance of each unit in the input layer or the hidden layers;

(vi) Bias which provides basic points for the activation functions.



**Fig 1.2.3.1 Deep Neural Network**

**1.2.4 Determine Person Location**

In determining the position of a person’s bounding box as well as the segment involved, each ground plane point is used to compare the ROI range.

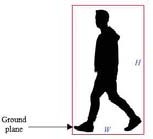


Fig 1.2.4.1 The ground plane of a human walking model.

Surveillance cameras are usually placed at high places as the overhead camera

especially to monitor a certain area e.g. high- risk area or areas of interest for an

organization. In this case, it is more suitable to compare the ground plane for the

detection box instead of using the center point value. calculate the center point of a

bounding box.

**Calculate the center point of a bounding box**

To measure the center point,of the bounding box for the detected person, midpoint

equation is used as in

C(x,y) = ( )

Each of the minimum and maximum value for the corresponding width,and height,

and of the bounding box will be used to calculate the center point of the bounding

box.

**Calculate distance between bounding box**

To measure the distance, C1(Xmin,Ymin) and C2(Xmax,Ymax), between each of the

detected person in the frame, distance equation is used as in

d(C1,C2) =

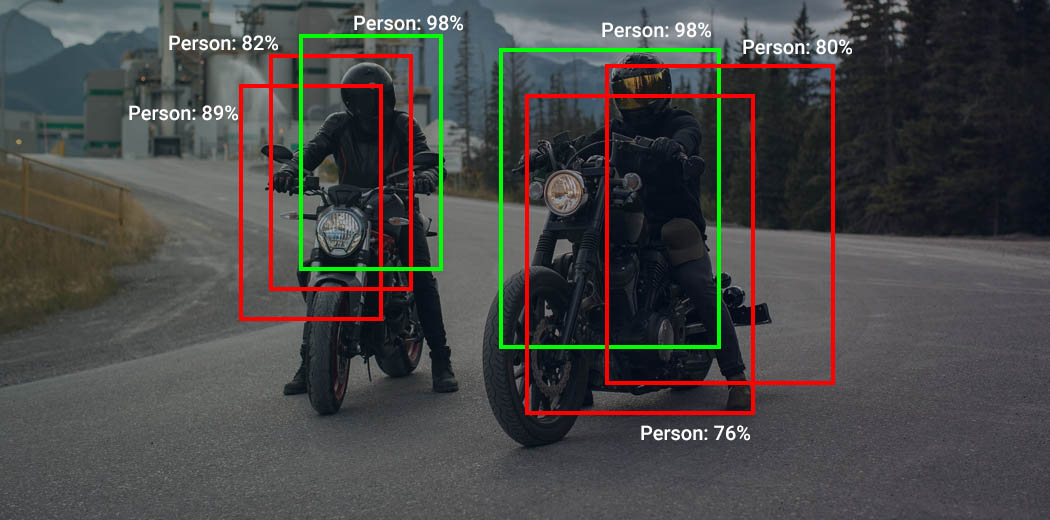
In this study, the center point of the bounding boxes is taken to determine between

two different locations of the bounding boxes. After getting the center points value,

the algorithm will calculate if the distance is lower or higher than 300 pixels

**1.2.5 Non-Maxima Suppression for object detection**

Non-max suppression is the final step of these object detection algorithms and is used to select the most appropriate bounding box for the object.



**Fig: 1.2.5 Non-maxima supression for a person**

The objects in the image can be of different sizes and shapes, and to capture each of these perfectly, the object detection algorithms create multiple bounding boxes. Ideally, for each object in the image, we must have a single bounding box. Something like the image on the green box. To select the best bounding box, from the multiple predicted bounding boxes, these object detection algorithms use non-max suppression. This technique is used to “suppress” the less likely bounding boxes and keep only the best one.

The purpose of non-max suppression is to select the best bounding box for an object and reject or “suppress” all other bounding boxes. The NMS takes two things into account

1. The objectiveness score is given by the model
2. The overlap or IOU of the bounding boxes

**1.3 Organization of Project**

The technique which is developed is taking input as a video and computes distances between detected persons in the frame and checks if any person is violating the specified distance and gives number of violations and if violations exceed than specified then it Alert mail to specified mail.

We have three modules in our project.

* Object detection.
* Social distance detection.
* Alert Mail.

**2. THEORETICAL ANALYSIS OF THE PROPOSED PROJECT**

**2.1 Requirements Gathering**

**2.1.1 Software Requirements**

Programming Language : Python 3.6

Graphical User Interface : Streamlit

Packages : Numpy, Imutils, Scipy, Opencv, smtplib

Tool : Pycharm

**2.1.2 Hardware Requirements**

Operating System: Windows 10

Processor : Intel Core i3-7th generation

CPU Speed : 2.30 GHz

Memory : 4 GB (RAM)

**2.2 Technologies Description**

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Advantages of Python:**

**1. Extensive Libraries**

Python downloads with an extensive library and it contains code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

**2. Extensible**

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

**3. Embeddable**

Complimentary to extensibility Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

**4. Improved Productivity**

The language simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. You can write less and get more things done.

**5. IOT Opportunities**

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.

**6. Simple and Easy**

When working with Java, you may have to create a class to print ’’Hello World‟. But in Python, just a print statement will do. It is also quite easy to learn, understand and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

**STREAMLIT**

**Streamlit** is one of the frameworks in Python which is used for building Web **APIs** very easily. **Streamlit is** an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science.

* Streamlit has a number of ways to add text to your app.
* Streamlit supports several popular data charting libraries like [Matplotlib, Altair, deck.gl, and more](https://docs.streamlit.io/en/stable/api.html#display-charts).
* With widgets, Streamlit allows us to bake interactivity directly into your apps with checkboxes, buttons, sliders, and more.

The goal is to use Streamlit to create an interactive app for your data or model and along the way to use Streamlit to review, debug, perfect, and share your code.

**DateTime:**

Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

The datetime classes are categorize into 6 main classes –

* [date](https://www.geeksforgeeks.org/python-datetime-module-with-examples/#date) – An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Its attributes are year, month and day.
* [time](https://www.geeksforgeeks.org/python-datetime-module-with-examples/#time) – An idealized time, independent of any particular day, assuming that every day has exactly 24\*60\*60 seconds. Its attributes are hour, minute, second, microsecond, and tzinfo.
* [datetime](https://www.geeksforgeeks.org/python-datetime-module-with-examples/#datetime) – Its a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo.
* [timedelta](https://www.geeksforgeeks.org/python-datetime-module-with-examples/#timedelta) – A duration expressing the difference between two date, time, or datetime instances to microsecond resolution.
* [tzinfo](https://www.geeksforgeeks.org/python-datetime-module-with-examples/#tzinfo) – It provides time zone information objects.
* [timezone](https://www.geeksforgeeks.org/python-datetime-module-with-examples/#timezone) – A class that implements the tzinfo abstract base class as a fixed offset from the UTC (New in version 3.2).

**Numpy**

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

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

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

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

**Scipy**

SciPy, a scientific library for Python is an open source, BSD-licensed library for mathematics, science and engineering. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. The main reason for building the SciPy library is that, it should work with NumPy arrays. It provides many user-friendly and efficient numerical practices such as routines for numerical integration and optimization.

SciPy is a collection of mathematical algorithms and convenience functions built on the Numpy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data. With SciPy an interactive Python session becomes a data-processing and system prototyping environment rivaling sytems such as MATLAB, IDL, Octave, R-Lab, and SciLab. The additional benefit of basing SciPy on Python is that this also makes a powerful programming language available for use in developing sophisticated programs and specialized applications. Scientific applications using SciPy benefit from the development of additional modules in numerous niche’s of the software landscape by developers across the world. Everything from parallel programming to web and data-base subroutines and classes have been made available to the Python programmer. All of this power is available in addition to the mathematical libraries in SciPy.

**Opencv**

OpenCV is the huge open-source library for the computer vision, machine learning,

and image processing and now it plays a major role in real-time operation which is

very important in today’s systems. By using it, one can process images and videos

to identify objects, faces, or even handwriting of a human. When it integrated with

various libraries, such as Numpuy, python is capable of processing the OpenCV

array structure for analysis. To Identify image pattern and its various features we

use vector space and perform mathematical operations on these features.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it’s free for both **academic** and **commercial** use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

**Imutils**

Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and both Python 2.7 and Python 3.

**Smtplib**

Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers.

Python provides **smtplib** module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon. The parameters are as follows-

* **host** − This is the host running your SMTP server. You can specify IP address of the host or a domain name like tutorialspoint.com. This is optional argument.
* **port** − If you are providing *host* argument, then you need to specify a port, where SMTP server is listening. Usually this port would be 25.
* **local\_hostname** − If your SMTP server is running on your local machine, then you can specify just *localhost* as of this option.

An SMTP object has an instance method called **sendmail**, which is typically used to do the work of mailing a message. It takes three parameters −

* The *sender* − A string with the address of the sender.
* The *receivers* − A list of strings, one for each recipient.
* The *message* − A message as a string formatted as specified in the various RFCs.

**SSL**

Requests verifies SSL certificates for HTTPS requests, just like a web browser. SSL Certificates are small data files that digitally bind a cryptographic key to an organization’s details. Often, an website with a SSL certificate is termed as secure website. By default, SSL verification is enabled, and Requests will throw a SSLError if it’s unable to verify the certificate.

**Pycharm**

PyCharm is the most popular IDE for Python, and includes great features such as excellent code completion and inspection with advanced debugger and support for web programming and various frameworks. PyCharm is created by Czech company, Jet brains which focusses on creating integrated development environment for various web development languages like JavaScript and PHP. It provides a wide range of essential tools for python developers, tightly integrated to create a convenient environment for productive python, web, and data science development.

Many programmers nowadays opt for [**Python**](https://www.edureka.co/blog/python-tutorial/) to build software applications with the concise, clean, and readable code base. They can even accelerate custom software application development by taking advantage of a number of integrated development environments (IDEs) for Python.

JetBrains has developed PyCharm as a cross-platform IDE for [**Python**](https://www.edureka.co/blog/python-tutorial/). In addition to supporting versions 2.x and 3.x of Python, PyCharm is also compatible with Windows, Linux, and macOS. At the same time, the tools and features provided by PyCharm help programmers to write a variety of software applications in Python quickly and efficiently.

The developers can even customize the PyCharm UI according to their specific needs and preferences. Also, they can extend the IDE by choosing from over 50 plug-ins to meet complex project requirements.

At present, the Python IDE is being used by large enterprises like Twitter, Pinterest, HP, Symantec, and Groupon.

## ****Features of PyCharm****

## ****Intelligent Code Editor****

PyCharm comes with a smart code editor that facilitates writing high-quality

Python code. It offers an enhanced level of code comprehension and readability

by means of distinct color schemes for keywords, classes, and functions, i.e.,

syntax and error highlighting.

In addition to offering the smart code completion feature, the code editor

generates instructions for completing the current code. Identifying errors and

issues is much more comfortable, along with linter integration and quick fixes.

### **Integrated Debugging and Testing**

An IDE comes with support for debugging and testing programs. To accomplish

the same, PyCharm features an integrated Python debugger and integrated unit

testing with line-by-line code coverage.

### **Refactoring**

The refactoring feature in PyCharm helps in improving the internal structure of a

Python program without affecting the external performance of the same. Making

changes to both local and global variables is efficient and fast.

The extract method is also there to split up extended classes and functions.

Other useful code refactoring features include:

* Introduce constant
* Introduce variable
* Pull up
* Push down
* Rename

### **Version Control Systems (VCSs) Integration**

In its simplicity, a version control system (VCS) keeps track of the changes made

to files, applications, and other sources of information. It can be considered as a

database of changes.

PyCharm provides a unified user interface for CVS, Git, Mercurial, Perforce, and

Subversion.

### **Project and Code Navigation**

The code navigation feature makes it much easier for developers to navigate to a

class, function, or file. It also helps in significantly cutting-down effort and time

required to edit and enhance the Python code. File structure views and specialized

project views are readily available.

The lens mode allows a developer to inspect and debug the entire Python source

code thoroughly. With code navigation, locating an element, variable, etc. is done

in almost no time. Developers can quickly jump between classes, files, and methods.

### **Advantages**

* A plethora of productive shortcuts
* Ability to view the entire Python source code with a single click
* Availability of an array of plugins
* Easy-to-use
* Excellent community support
* Facilitates faster code development
* More powerful, commercial version available
* Straightforward installation process

### **Disadvantages**

* Costly paid version
* May pose issues when trying to fixing tools like venv
* Not suitable for Python beginners
* Resource-intensive application, i.e., requires plenty of memory and storage space

**3. DESIGN**

**3.1 Introduction**

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

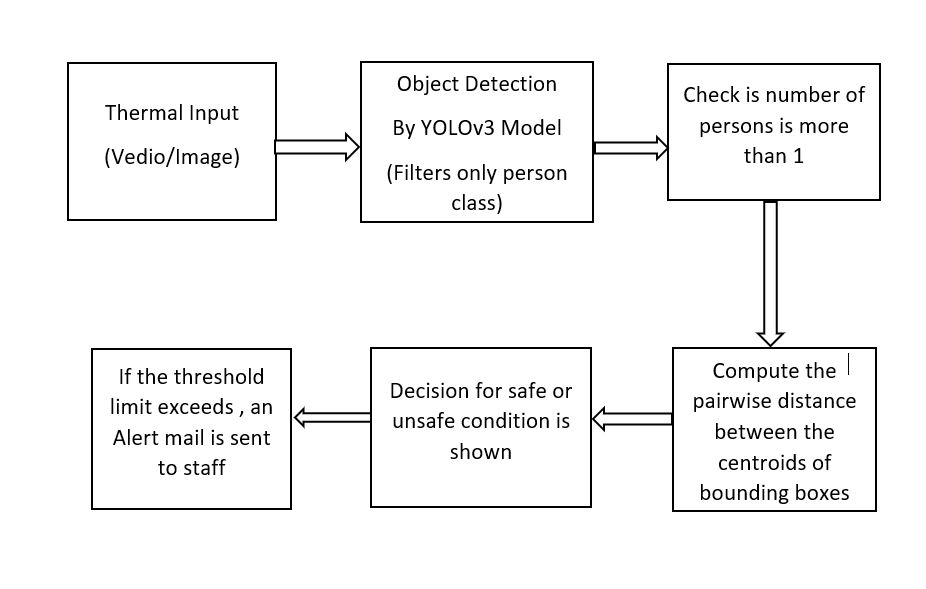
The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

**3.2 Architecture Diagram**

Web applications are by nature distributed applications, meaning that they are programs that run on more than one computer and communicate through network or server. Specifically, web applications are accessed with a web browser and are popular because of the ease of using the browser as a user client. For the enterprise, software on potentially thousands of client computers is a key reason for their popularity. Web applications are used for web mail, online retail sales, discussion boards, weblogs, online banking, and more. One web application can be accessed and used by millions of people.

Like desktop applications, web applications are made up of many parts and often contain mini programs and some of which have user interfaces. In addition, web applications frequently require an additional markup or scripting language, such as HTML, CSS, or JavaScript programming language. Also, many applications use only the Python programming language, which is ideal because of its versatility.



**Fig 3.2.1: Architecture Diagram**

**Social Distance Detector Steps**

This section discusses the essential steps which are attempted to establish a workflow for monitoring social distancing on images as seen in fig 3.2.

1. Prepare the images or streaming a video from a camera which contains people.

2. Applying the deep learning object detector to detect people in images or video streams.

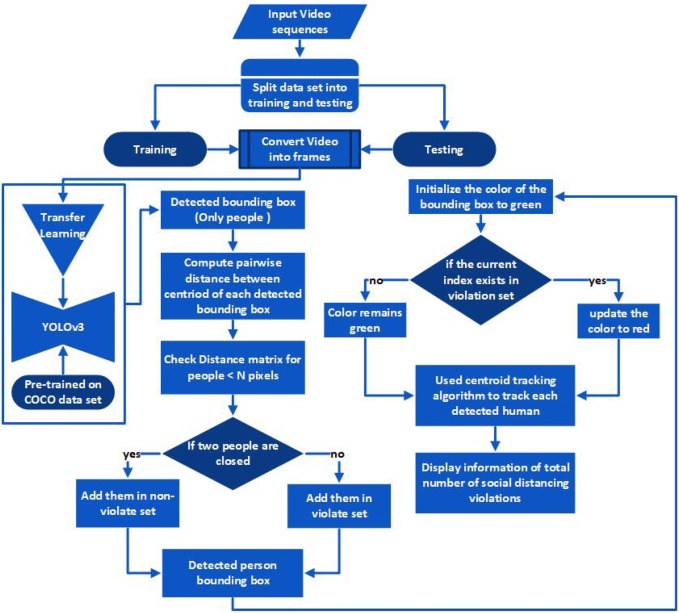
3. Check the number of persons that are in the images or video stream.

4. Compute the distance between the centroid of the bounding boxes which are enclosed to the detected people.

5. Finally, the algorithm will decide for safe or unsafe social distancing based on the number of persons and the measured distance between the centroid of bounding boxes.

6. And if the given threshold limit value is reached, an alert mail is sent to the monitoring staff.

**3.2.2 FLOW DIAGRAM :**

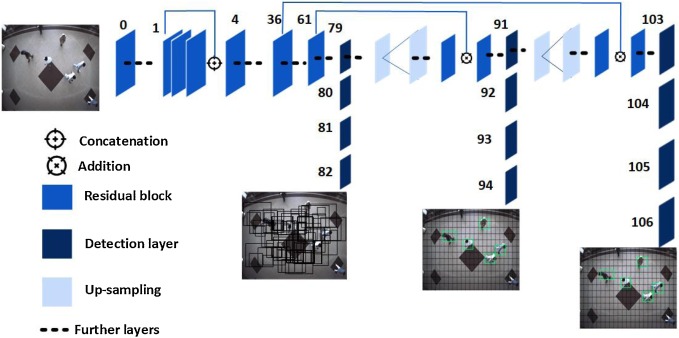
****

**Fig:3.2.2 Flow diagram**

The flow diagram of the framework is shown above. Due to the best performance results for generic object detection, in this work, YOLOv3 is used. The model used single-stage network architecture to estimate the bounding boxes and class probabilities. The model was originally trained on the COCO (Common objects in context) data set .

After detection, the bounding box information, mainly centroid information, is used to compute each bounding box centroid distance. We used Euclidean distance and calculated the distance between each detected bounding box of peoples. Following computing centroid distance, a predefined threshold is used to check either the distance among any two bounding box centroids is less than the configured number of pixels or not. If two people are close to each other and the distance value violates the minimum social distance threshold. The bounding box information is stored in a violation set, and the color of the bounding box is updated/changed to red. A centroid tracking algorithm is adopted for tracking so that it helps in tracking of those people who violate/breach the social distancing threshold. At the output, the model displays the information about the total number of social distancing violations along with detected people bounding boxes and centroids.

In this work, YOLOv3 is used for human detection as it improves predictive accuracy, particularly for small-scale objects. The main advantage is that it has adjusted network structure for multi-scale object detection. Furthermore, for object classification, it uses various independent logistic rather than softmax. The model’s overall architecture is presented above, it can be seen that feature learning is performed using the convolutional layers, also called Residual Blocks. The blocks are made up of many convolutional layers and skip connections.



**Fig 3.2.3: Single-Stage Network**

The architecture shown in above used a single-stage network for the entire input image to predict the bounding box and class probability of detected objects. For feature extraction, the architecture utilizes convolution layers, and for class prediction, fully connected layers are used. During human identification, as seen in above image, the input frame is divided into a region of S×S, also called grid cells. These cells are related to bounding box estimation and class probabilities. It predicts the probability of whether the center of the person bounding box is in the grid cell or not:

Conf(p) = Pr(p) \* IOU(pred,actual) -------(1)

In Eq. [(1)](https://www.sciencedirect.com/science/article/pii/S2210670720307897" \l "eq0005), Pr(p) indicates that whether the person present is in the detected bounding box or not. The value of Pr(p) is 1 for yes and 0 for not. IoU(pred,actual) determines the Intersection Over Union of the actual and predicted bounding box. It is defined as

IoU(pred,actual)=area BoxT∩BoxP/ BoxT∪BoxP----------(2)

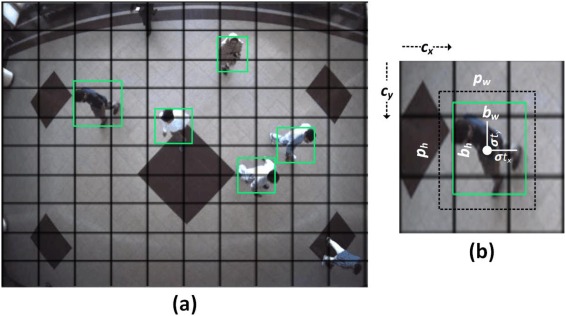
where the ground truth box (actual) manually labeled in the training data set represented with BoxT, and the predicted bounding box is displayed as BoxP. area presents the area of intersection. An acceptable area is predicted and decided for each detected person in the input frame. The confidence value is applied after prediction to achieve the optimal bounding box. For each predicted bounding box, h,w,x,y are estimated, where bounding box coordinates are defined by x,y, and width and height are determined by w,h. The model produces the following predicted bounding box values

bx = σ(tx) + cx

by = σ(ty) + cy

bw = pwet w

bh = pwht

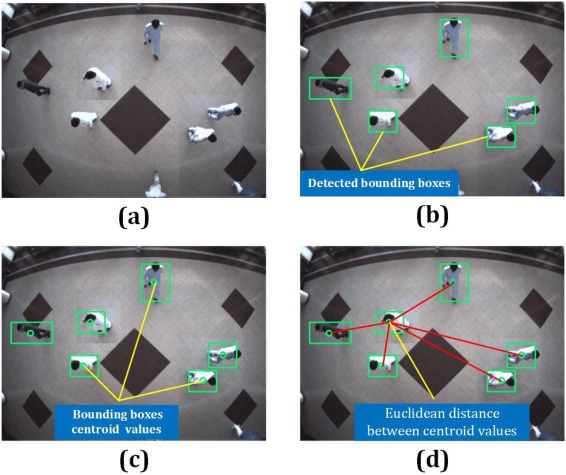


**Fig. 3.2.4. Detected coordinates of person bounding box.**

In above, bx,by,bw,bh are predicted coordinate bounding boxes, where the coordinates’ center is represented as x,y and width and height with w,h. tw,th,tx,ty, defined the network output and cx,cy are used to correspond the top-left coordinates of the grid cell as shown in fig, while the pw and ph are width and height of anchors.

A threshold value is defined that process the high confidence values and discards the low confidence values. Using non-maximal suppression, the final location parameters are derived for the detected bounding box. At last, loss function is calculated, for detected bounding box. The given loss function is the sum of three functions, i.e., regression, classification, and confidence. At each grid cell, if the object is detected, then the classification loss is computed as the squared error of the conditional class probabilities are calculated .

After detecting people in video frames, in the next step, the centroid of each detected person bounding boxes shown as green boxes are used for distance calculation. The detected bounding box coordinates (x,y) are used to compute the bounding box's centroid, demonstrates accepting a set of bounding box coordinates and computing the centroid. After computing, centroid, a unique ID is assigned to each detected bounding box. In the next step, we measure the distance between each detected centroid using Euclidean distance. For every subsequent frame in the video stream, we firstly compute bounding box centroids; and then calculate the distance (highlighted with red lines) between each pair of detected bounding box centroids. The information of each centroid is stored in the form of a list. Based on distance values, a threshold is defined to check if any two people are less than N pixels apart or not. If the distance violates the minimum social distance set or two people are too close, then the information is added into the violation set. The bounding box color is initialized as green. The information is checked in the violation set; if the current index exists violation set, the color is updated to red. Furthermore, the centroid tracking algorithm is used to track the detected people in the video sequence. The tracking algorithm also helps to keep track of people who are violating the social distance threshold. At the output, the model displays information about the total number of social distancing violations. And if the limit (i.e, minimum violations count) exceeds, then an alert mail with message “Social distancing violations exceeded” is sent to the specified monitoring staff members.



**Fig: 3.2.5** (a) Input image, (b) detected person bounding boxes using deep learning algorithm, (c) compute the centroid of each detected bounding box, and (d) finally, the distance between each pair of the centroid is determined

**3.3 UML Diagrams**

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

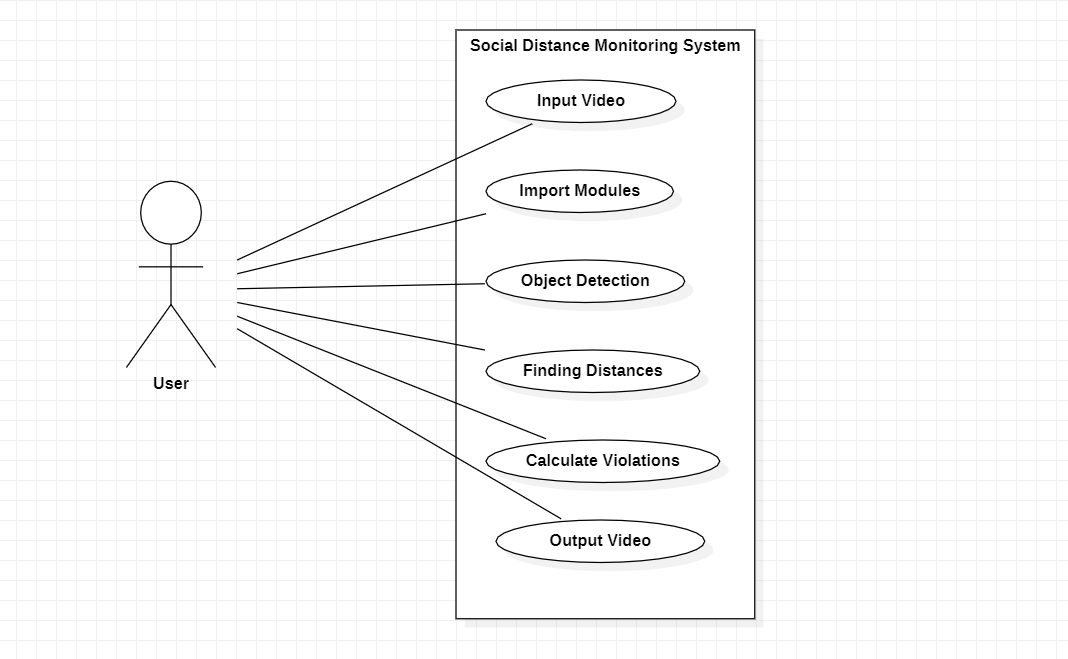
**3.3.1 Use Case Diagram**

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating.

Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consist of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

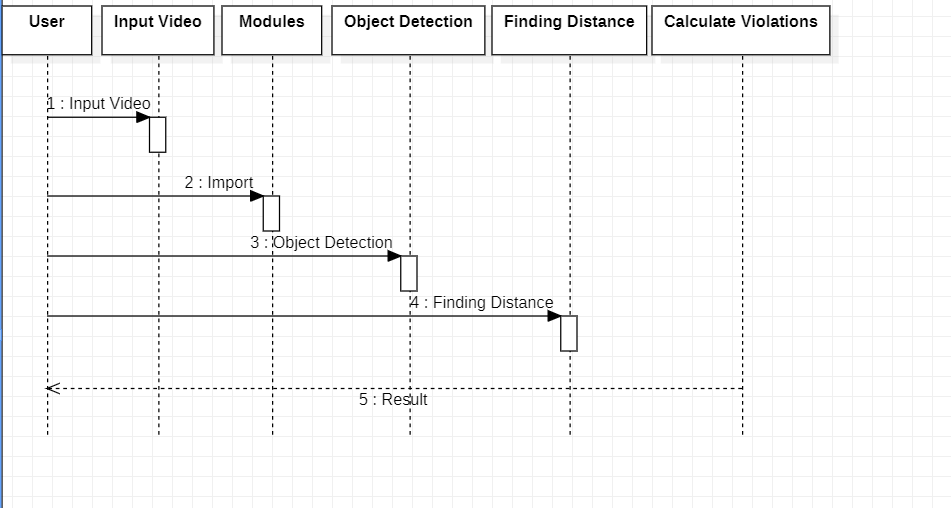
Hence to model the entire system, a number of use case diagrams are used.

****

**Fig 3.3.1: Use Case Diagram**

**3.3.2 Sequence Diagram**

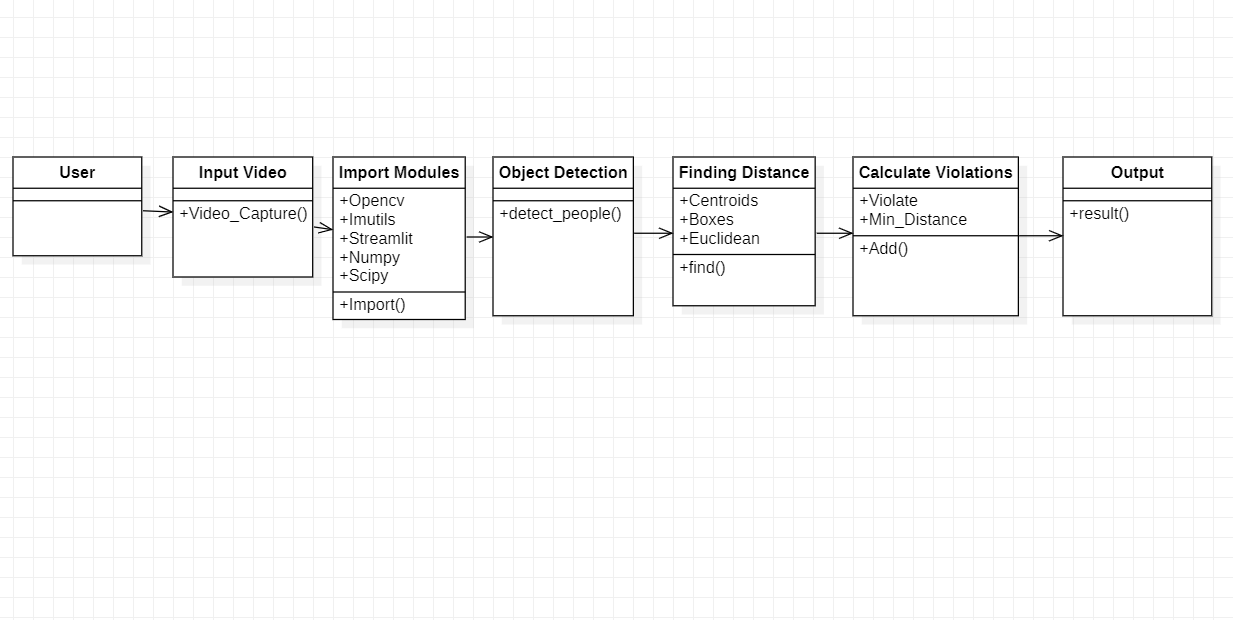
Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.



**Fig 3.3.2: Sequence Diagram**

**3.3.3 Class Diagram**

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.



**Fig 3.3.3: Class Diagram**

**4. IMPLEMENTATION**

**4.1 Coding**

**Social\_distance\_detector\_config.py**

MODEL\_PATH = **"yolo-coco"**

USE\_GPU = False

Threshold = 15

ALERT = True  
MAIL = **'radha@gmail.com'**

**Detection.py**

from .social\_distancing\_config import NMS\_THRESH  
from .social\_distancing\_config import MIN\_CONF  
import numpy as np  
import cv2  
  
def detect\_people(frame, net, ln, personIdx=0):  
 (H, W) = frame.shape[:2]  
 results = []  
  
 blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416),

swapRB=True, crop=False)  
 net.setInput(blob)  
 layerOutputs = net.forward(ln)  
  
 boxes = []  
 centroids = []  
 confidences = []  
 for output in layerOutputs:  
 for detection in output:  
 scores = detection[5:]  
 classID = np.argmax(scores)  
 confidence = scores[classID]  
 if classID == personIdx and confidence > MIN\_CONF:  
 box = detection[0:4] \* np.array([W, H, W, H])  
 (centerX, centerY, width, height) = box.astype(**"int"**)  
 x = int(centerX - (width / 2))  
 y = int(centerY - (height / 2))  
 boxes.append([x, y, int(width), int(height)])  
 centroids.append((centerX, centerY))  
 confidences.append(float(confidence))  
  
 idxs = cv2.dnn.NMSBoxes(boxes, confidences, MIN\_CONF, NMS\_THRESH) if len(idxs) > 0:  
 for i in idxs.flatten():  
 (x, y) = (boxes[i][0], boxes[i][1])  
 (w, h) = (boxes[i][2], boxes[i][3])  
 r = (confidences[i], (x, y, x + w, y + h), centroids[i])  
 results.append(r) *#updating results list* return results

**Mailer.py**

import smtplib, ssl  
  
class Mailer:

def \_\_init\_\_(self):

self.EMAIL = **"abc123@gmail.com"**

self.PASS = **"ncfdjpnvfyitcole"** self.PORT = 465  
 self.server = smtplib.SMTP\_SSL(**'smtp.gmail.com'**, self.PORT)

def send(self, mail):  
 self.server = smtplib.SMTP\_SSL(**'smtp.gmail.com'**, self.PORT)  
 self.server.login(self.EMAIL, self.PASS)

SUBJECT = **'ALERT!'** TEXT = **f'Social distancing violations exceeded!'** message = **'Subject: {}**\n\n**{}'**.format(SUBJECT, TEXT)

self.server.sendmail(self.EMAIL, mail, message)  
 self.server.quit()

**AlertCode.py**

from distancein import social\_distancing\_config as config  
from distancein.detection import detect\_people  
from scipy.spatial import distance as dist  
import numpy as np  
import imutils  
import cv2  
import os  
import streamlit as st  
import datetime  
import time  
  
  
st.title(**"Social Distancing Detector"**)  
st.subheader(**'A GUI Based Social Distancing Detector'**)  
  
MIN\_CONF = st.slider(**'Minimum probability To Filter Weak Detections'**, 0.0, 1.0, 0.3)  
NMS\_THRESH = st.slider(**'Non-Maxima suppression Threshold'**, 0.0, 1.0, 0.3)  
  
st.subheader(**'Test Demo Video Or Try Live Detection'**)  
option = st.selectbox(**'Choose your option'**,  
 (**'Demo1'**, **'Demo2'**,**'Demo3'**,**'Demo4'**,**'Demo5'**, **'Try Live Detection Using Webcam'**))  
  
MIN\_CONF = float(MIN\_CONF)  
NMS\_THRESH = float(NMS\_THRESH)  
  
MIN\_DISTANCE = 50  
  
labelsPath = os.path.sep.join([config.MODEL\_PATH, **"coco.names"**])  
LABELS = open(labelsPath).read().strip().split(**"**\n**"**)  
  
weightsPath = os.path.sep.join([config.MODEL\_PATH, **"yolov3.weights"**])  
configPath = os.path.sep.join([config.MODEL\_PATH, **"yolov3.cfg"**])  
  
net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)  
  
ln = net.getLayerNames()  
ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]  
  
if st.button(**'Start'**):  
  
 st.info(**"[INFO] loading YOLO from disk..."**)  
 st.info(**"[INFO] accessing video stream..."**)  
 if option == **"Demo1"**:  
 vs = cv2.VideoCapture(**'input.mp4'**)  
 elif option == **"Demo2"**:  
 vs = cv2.VideoCapture(**'input2.mp4'**)  
 elif option == **"Demo3"**:  
 vs = cv2.VideoCapture(**'input3.mp4'**)  
 elif option == **"Demo4"**:  
 vs = cv2.VideoCapture(**'input4.mp4'**)  
 elif option == **"Demo5"**:  
 vs = cv2.VideoCapture(**'input5.mp4'**)  
 else:  
 vs = cv2.VideoCapture(0)  
 writer = None  
  
 image\_placeholder = st.empty()  
 time1 = time.time()  
 while True:  
  
 (grabbed, frame) = vs.read()  
  
 if not grabbed:  
 break  
  
 frame = imutils.resize(frame, width=700)  
 results = detect\_people(frame, net, ln,  
 personIdx=LABELS.index(**"person"**))  
  
 violate = set()  
  
 if len(results) >= 2:  
  
 centroids = np.array([r[2] for r in results])  
 D = dist.cdist(centroids, centroids, metric=**"euclidean"**)  
  
 for i in range(0, D.shape[0]):  
 for j in range(i + 1, D.shape[1]):  
  
 if D[i, j] < MIN\_DISTANCE:  
  
 violate.add(i)  
 violate.add(j)  
  
 for (i, (prob, bbox, centroid)) in enumerate(results):  
  
 (startX, startY, endX, endY) = bbox  
 (cX, cY) = centroid  
 color = (0, 255, 0)  
  
 if i in violate:  
 color = (0, 0, 255)  
  
 cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)  
 cv2.circle(frame, (cX, cY), 5, color, 1)  
  
 font = cv2.FONT\_HERSHEY\_SIMPLEX

Threshold = **"Threshold limit: {}"**.format(config.Threshold)  
 cv2.putText(frame, Threshold, (470, frame.shape[0] - 50),  
 cv2.FONT\_HERSHEY\_SIMPLEX, 0.60, (255, 0, 0), 2)  
 text = **"Social Distancing Violations: {}"**.format(len(serious))  
 cv2.putText(frame, text, (10, frame.shape[0] - 25),  
 cv2.FONT\_HERSHEY\_SIMPLEX, 0.85, (0, 0, 255), 3)

if len(serious) >= config.Threshold:  
 cv2.putText(frame, **"-ALERT: Violations over limit-"**, (10, frame.shape[0] - 80),  
 cv2.FONT\_HERSHEY\_COMPLEX, 0.60, (0, 0, 255), 2)  
 if config.ALERT:  
  
 st.info(**"[INFO] Sending mail..."**)  
  
 Mailer().send(config.MAIL)  
 st.info(**'[INFO] Mail sent'**)

display = 1  
 if display > 0:  
  
 image\_placeholder.image(  
 frame, caption=**'Live Social Distancing Monitor Running..!'**, channels=**"BGR"**)  
  
 if writer is not None:  
 writer.write(frame)  
 time2 = time.time()  
 st.info(**'Time Taken: '**,(time2-time1)/60)  
  
st.success(**"STAY HOME STAY SAFE"**)

**4.2 Testing**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of software as a system element and attendant costs associated with a software failure are motivating factors for we planned, through testing. Testing is the process of executing a program with the intent of finding an error. The design of tests for software and other engineered products can be as challenging as the initial design of the product itself.

There of basically two types of testing approaches.

One is Black-Box testing – the specified function that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operated.

The other is White-Box testing – knowing the internal workings of the product ,tests can be conducted to ensure that the internal operation of the product performs according to specifications and all internal components have been adequately exercised.

White box and Black box testing methods have been used to test this package. The entire loop constructs have been tested for their boundary and intermediate conditions. The test data was designed with a view to check for all the conditions and logical decisions. Error handling has been taken care of by the use of exception handlers.

**4.2.1 Testing Strategies**

Testing is a set of activities that can be planned in advanced and conducted systematically. A strategy for software testing must accommodation low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

Software testing is one element of verification and validation. Verification refers to the set of activities that ensure that software correctly implements as specific function. Validation refers to a different set of activities that ensure that the software that has been built is traceable to customer requirements.

The main objective of software is testing to uncover errors. To fulfill this objective, a series of test steps unit, integration, validation and system tests are planned and executed. Each test step is accomplished through a series of systematic test technique that assist in the design of test cases. With each testing step, the level of abstraction with which software is considered is broadened.

Testing is the only way to assure the quality of software and it is an umbrella activity rather than a separate phase. This is an activity to be performed in parallel with the software effort and one that consists of its own phases of analysis, design, implementation, execution and maintenance.

UNIT TESTING:

This testing method considers a module as single unit and checks the unit at interfaces and communicates with other modules rather than getting into details at statement level. Here the module will be treated as a black box, which will take some input and generate output. Outputs for a given set of input combination are pre-calculated and are generated by the module.

SYSTEM TESTING:

Here all the pre tested individual modules will be assembled to create the larger system and tests are carried out at system level to make sure that all modules are working in synchronous with each other. This testing methodology helps in making sure that all modules which are running perfectly when checked individually are also running in cohesion with other modules. For this testing we create test cases to check all modules once and then generated test combinations of test paths through out the system to make sure that no path is making its way into chaos.

INTEGRATED TESTING

Testing is a major quality control measure employed during software development. Its basic function is to detect errors. Sub functions when combined may not produce than it is desired. Global data structures can represent the problems. Integrated testing is a systematic technique for constructing the program structure while conducting the tests. To uncover errors that are associated with interfacing the objective is to make unit test modules and built a program structure that has been detected by design. In a non - incremental integration all the modules are combined in advance and the program is tested as a whole. Here errors will appear in an end less loop function. In incremental testing the program is constructed and tested in small segments where the errors are isolated and corrected.

Different incremental integration strategies are top – down integration, bottom – up integration, regression testing.

REGRESSION TESTING

Each time a new module is added as a part of integration as the software changes. Regression testing is an actually that helps to ensure changes that do not introduce unintended behavior as additional errors.

Regression testing maybe conducted manually by executing a subset of all test cases or using automated capture play back tools enables the software engineer to capture the test case and results for subsequent playback and compression. The regression suit contains different classes of test cases.

A representative sample to tests that will exercise all software functions.

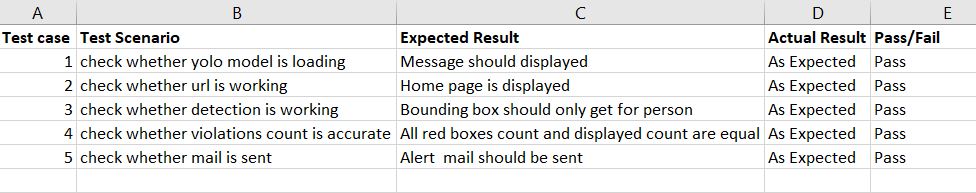
Additional tests that focus on software functions that are likely to be affected by the change.

ACCEPTANCE TESTING

When that user fined no major problems with its accuracy, the system passers through

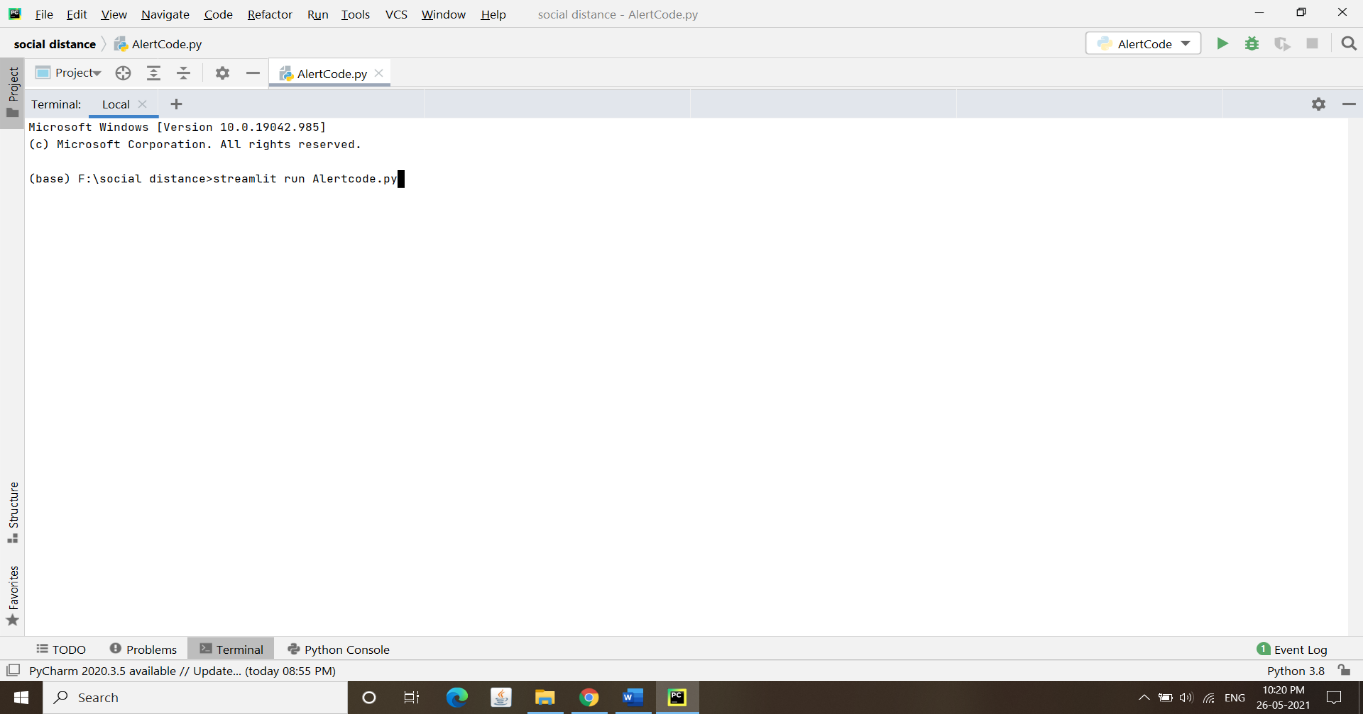
a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

**4.3 TEST CASES**

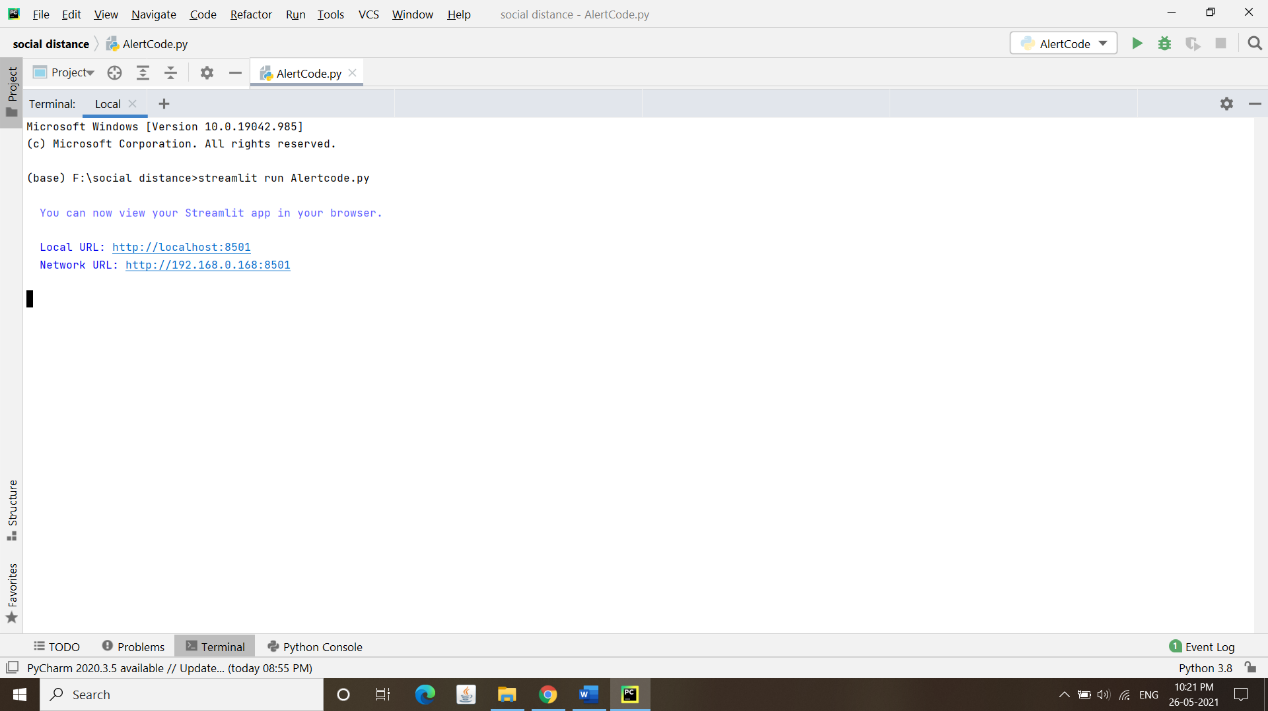
****

**Fig 4.3: Test cases**

**4.4 INPUT SCREENSHOTS**

****

**Fig 4.4.1: Command to run project**

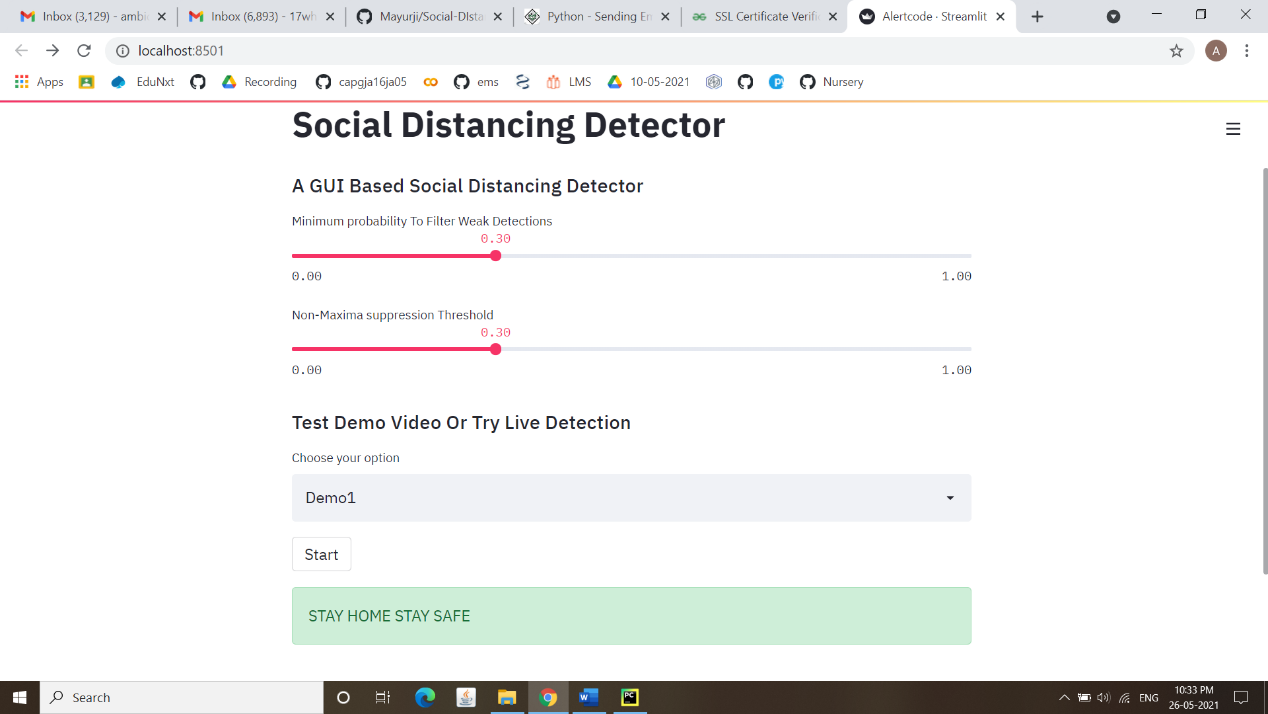
****

**Fig 4.4.2: Website URL**

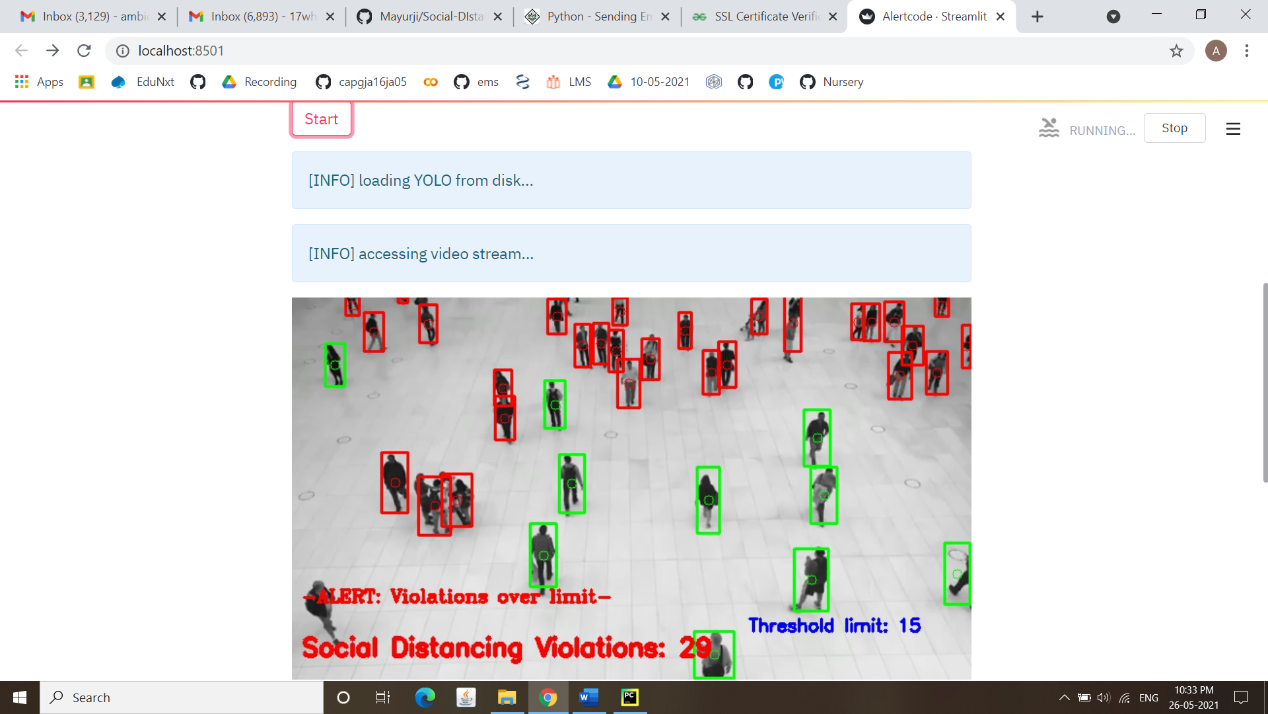
****

**Fig 4.4.3: Frame of Input video**

**4.5 OUTPUT SCREENSHOTS**

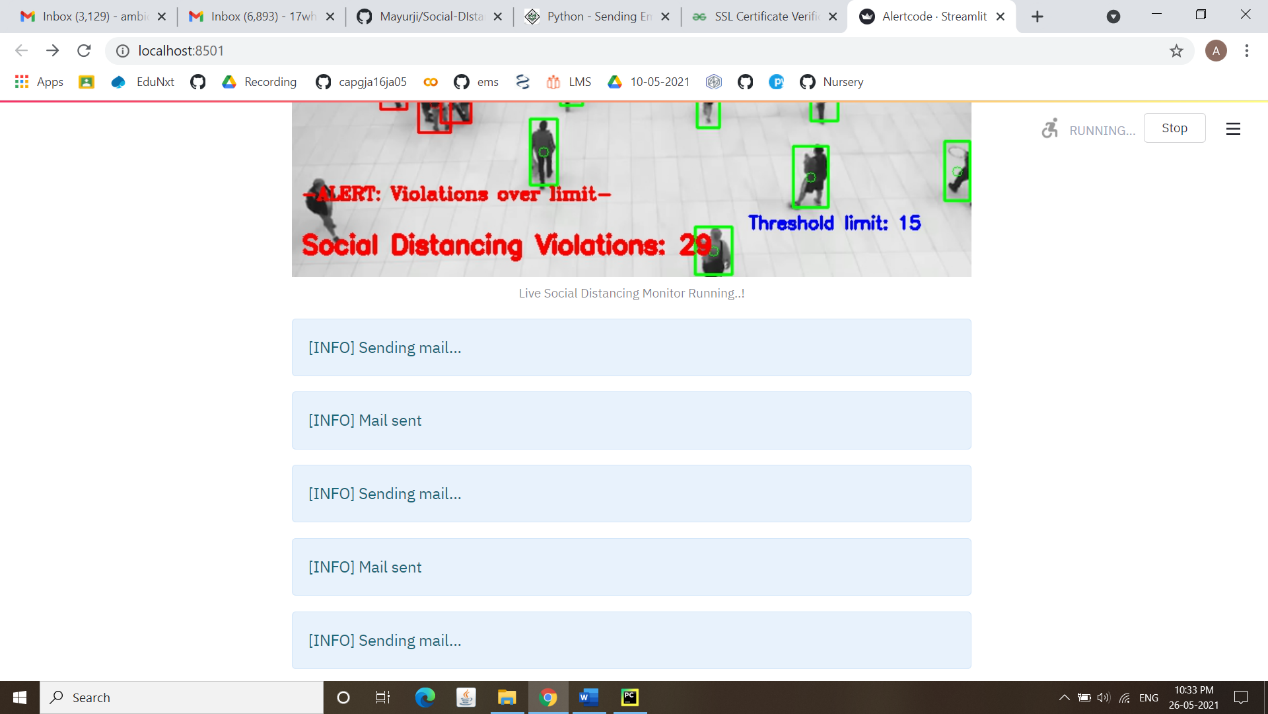
****

**Fig 4.5.1: A Web Page where we give our input video**

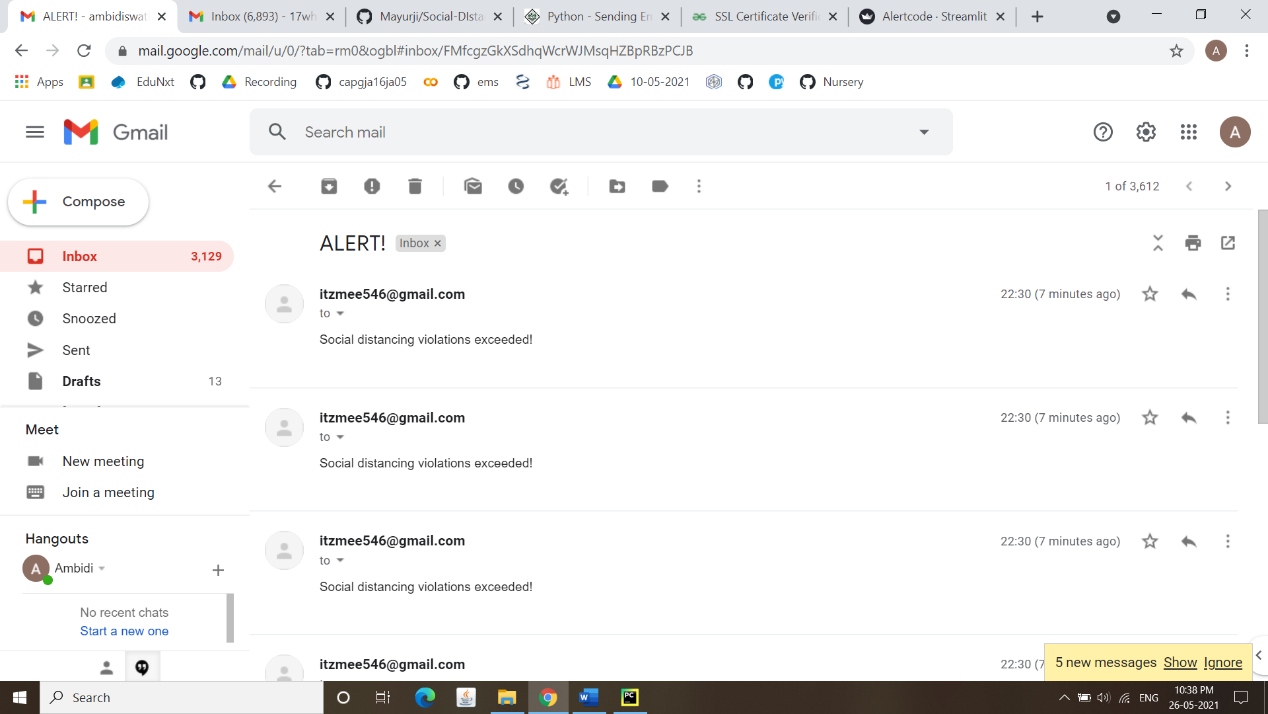
****

**Fig 4.5.2: A picture showing that the model started detection**

**\**

****

**Fig 4.5.3: An information displaying that mails are sent when violation limit is reached**

****

**Fig 4.5.4: Mails sent to the respective monitoring staff members**

**5.CONCLUSION AND FUTURE SCOPE**

The emerging trends and the availability of intelligent technologies make us to develop new models that help to satisfy the needs of emerging world. So we have developed a novel social distancing detector which can possibly contribute to public healthcare. The model proposes an efficient real-time deep learning based framework to automate the process of monitoring the social distancing via object detection, tracking and alerting approaches, where each individual is identified in the real-time with the help of bounding boxes. The number of violations is confirmed by computing the number of groups formed and violation index term computed as the ratio of the number of people to the number of groups. Finally, if the minimum violations count is reached, an alerting mail to sent to monitors.

This system works very effectively and efficiently in identifying the social distancing between the people and generating the alert that can be monitored. The model can be extended by performing GPS location tracking which helps to identify the area where the crowd is violating the norms of social distancing

**6.REFERENCES**

[1] Advice for the public on COVID-19—World Health Organization. WHO [www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public](http://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public)

[2] https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/

[3]https://www.pyimagesearch.com/2020/02/10/openc v-dnn-with-nvidia-gpus-1549-faster-yolo-ssd-and-mask-rcnn/

[4] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7603992/

[5] World Health Organization, “Coronavirus Disease 2019,” Coronavirus disease (COVID-19) pandemic, 2020. https://www.who.int/emergencies/diseases/novel-coronavirus-2019 (accessed Jun. 19, 2020).

[6]M.Ministryof Health, “COVID-19 MALAYSIA,” Http://Covid19.Moh.Gov.My/, 2020. <http://covid-19.moh.gov.my/>.

[7]https://www.analyticsvidhya.com/blog/2018/12/prac tical-guide-object-detection-yolo-framewor-python/

[8]https://www.coursera.org/projects/real-time-object-detection-yolo