A

Mini Project Report

on

**AI- Enabled Surveillance and Analysis**

*Submitted in partial fulfillment of the requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

IN

**COMPUTER SCIENCE AND ENGINEERING**

**BY**

**Mr. Ashok Kumar MTech (Ph.D.)**

**Asst. Professor**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SPHOORTHY ENGINEERING COLLEGE**

**(AFFILIATED TO JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD)**

**NADARGUL (V), BALAPUR (M), RANGAREDDDY (D) – 501 510,**

**TELANGANA, INDIA**

**2018-2022**

**DECLARATION**

We the undersigned, declare that the mini project title “**AI- ENABLED SURVEILLANCE AND ANALYSIS**” carried out at “SPHOORTHY ENGINEERING COLLEGE” is original and isbeing submitted to the Department of COMPUTER SCIENCE AND ENGINEERING, Sphoorthy Engineering College, Hyderabad towards partial fulfillment for the award of Bachelor of Technology.

We declare that the result embodied in the Mini Project work has not been submitted to any other University or Institute for the award of any Degree or Diploma.

**Date:**

**Place: Hyderabad**

**Mr. GADI SAI CHARAN (18N81A05N7)**

**Mr. RAPOLU MANISH (18N81A05P1)**

**Mr. C. ROHITH (18N81A05P5)**

**Ms. ZEENATH AHMED (18N81A05L0)**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING CERTIFICATE**

This is to certify that project entitled “**AI- ENABLED SURVEILLANCE AND ANALYSIS**” is a bonafide work carried out by **Mr. GADI SAI CHARAN (18N81A05N7), Mr. RAPOLU MANISH (18N81A05P1) , Mr. C. ROHITH (18N81A05P5), Ms. ZEENATH AHMED (18N81A05L0) .** in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering, Sphoorthy Engineering College, Hyderabad under the supervision or guidance of “**Mr, Ashok Kumar**”. The result embodied in the Mini Project Work has not been ­­submitted to any other University or Institute for the award of any Degree or Diploma.

**INTERNAL GUIDE** **HEAD OF DEPARTMENT**



**EXTERNAL EXAMINER** **PRINCIPAL**



**ACKNOWLEDGEMENT**

We express our deep sense of gratitude to Our Project Guide **Mr. Ashok Kumar** Department of Computer Science & Engineering, Sphoorthy Engineering College, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad, for her inspiring guidance, consistent encouragement, constructive criticism and helpful suggestions during the entire course of our research work.

We express our sincere thanks to **Mr. P. Ram Mohan Rao,** Associate Professor & Head of the Department, Department of Computer Science & Engineering, Sphoorthy Engineering College, Nadergul (V), Balapur (M), Ranga Reddy (D) for his encouragement which helped us to complete our Project work.

We deem it a great privilege to express our profound gratitude and sincere thanks to **Mr. Chalama Reddy,** Chairman, **Mr. Jagan Mohan Reddy,** Secretary, **Prof. Subrahmanyam,** Principal**, Prof. MVS Ram Prasad,** Director, Sphoorthy Engineering College, Nadergul (V), Balapur (M), Ranga Reddy (D), for their moral support and help in the completion of our research work.

**ABSTRACT**

The idea is to develop “Artificially Intelligent Machine” for the problems faced while identifying the previous gathering chain of the newly identified patient. This helps in identifying the people who were potentially exposed with the patient. Preparing a “Artificially Intelligent Analysis test” to enquire about the symptoms of the person to take him under quarantine.

The real-time pandemic situation caused due to COVID\_19 spreads through physical contact. To stop it from spreading, we are maintaining self-isolation. Regardless of this, some people are violating the rules against “SOCIAL DISTANCING”.

To handle situations like this, we are taking it as a responsibility and developing technical solutions for “Smart Monitoring” and “Smart analysis”. By evaluating the data from the monitoring system, we intend people who violated the rule must take the Smart analysis test to enquire about their symptoms.

**INDEX**

**Topics of content Page no’s**

**DECLARATION 2**

**CERTIFICATE 3**

**ACKNOWLEDGMENT 4**

**ABSTRACT** **5**

**CHAPTER 1- INTRODUCTION 9**

**CHAPTER 2- LITERATURE SURVEY 10**

**2.1** Existing System

**2.2** Proposed System

**CHAPTER 3- SYSTEM REQUIREMENTS 11**

**3.1** Hardware requirements

**3.2** Software requirements

**Topics of content Page no’s**

**CHAPTER 4- SYSTEM DESIGN**

**4.1** UML

**4.1.1** Data diagrams **12**

**4.1.2** Flowcharts **14**

**4.1.3** Feasibility **15**

**4.1.4** Class Diagrams **16**

**CHAPTER 5- SOFTWARE ENVIRONMENTS**

**5.1** Technology used **17**

**5.2** Tools **25**

**5.3** Software environments **28**

**CHAPTER-6 IMPLEMENTATION**

**6.1** Surveillance system **29**

**6.2** Analysis system **37**

**Topics of content Page no’s**

**CHAPTER-7 SCREENSHOTS**

**7.1** CMD Execution **40**

**7.2** Live Monitoring **43**

**7.3** Prototype of the analysis software **48**

**CHAPTER-8 TESTING**

**8.1** Black Box Testing **49**

**8.2** White Box Testing **50**

**CHAPTER-9 CONCLUSION 51**

**CHAPTER 1 – INTRODUCTION**

The World had faced a pandemic in 2019. Covid-19 forced everyone to stay at home. Following this, we had social distancing practices implemented in public places. It became a standard practice throughout the World--maintaining 6 feet distance from another person while also wearing a mask became our routine.

Busy places like Airports, Hospitals, Metro Stations implemented strict social distancing. These rules became a key factor in fighting the virus. Now, social distancing is such an important practice; that violating it--can, in some countries, impose fine on the person. That said, making sure that everyone maintains social distancing is a great task. It requires a lot of personnel to monitor people. This is a problem

To handle situations like these, we have taken this as a responsibility on us and developed technical solutions for “Smart Monitoring” and “Smart analysis”.

An Artificially Intelligent Smart surveillance system which monitors, records only where the social distancing is violated without supervision of anyone will help the country in a large scale. Which can be easily installed in security cameras to monitor the people who are violating the rules against “Social distancing”.

By evaluating the data from the monitoring system, we intend people who violated the rules will have to take the Smart analysis test to understand more about their symptoms This would help us in evaluating the symptoms and problems that people developed during or post involvement of a crowd gathering (i.e., Violation)

**CHAPTER 2 - LITERATURE SURVEY**

**2.1 Existing System**

Till now our World has not had a major Lockdown since The World War 2 and the pandemic had somehow forced us to stay at home. Social distancing at public places is unheard of. But we can only identify based on patient lists and passengers list.

**2.2 Proposed System**

To overcome this problem of having someone to monitor every time there is a violation of social- distancing, we have built an AI- Enabled Surveillance and Analysis software.

This software records the Instances where there is a violation of Social- distancing.

This input for the software can be an image, a pre-recorded video, or Live footage.

By using the recognition of the face in the video we are able to calculate the Euclidian distance from both the centers of the recognized face

The best part of these types of surveillance systems is that it records only that part of the video where the people have violated social distancing.

**CHAPTER 3-SYSTEM REQUIREMENTS**

**3.1 Hardware Requirements**

1. RAM-4GB
2. Microprocessor 1 GHz
3. Security cameras in case of live monitoring

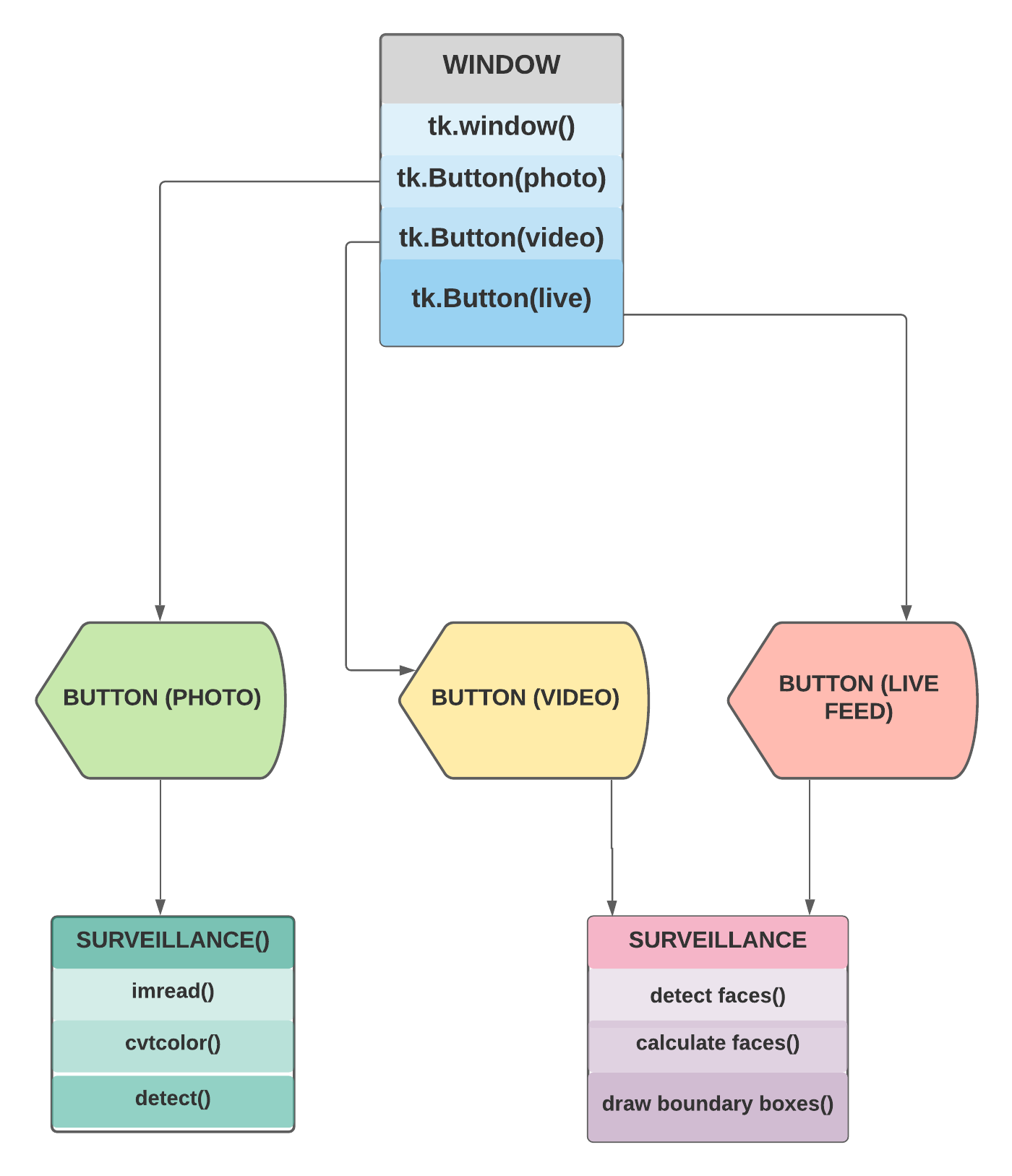
**3.2 Software Requirements**

|  |  |  |
| --- | --- | --- |
| ID | Used Need | Source |
| 1 | An Intelligent system that can identify people whoever are violating the social distancing without any human. intervention | AI enabled Surveillance System |
| 2 | The system should take photos, videos or live feed as an input i.e., should be compatible with all 3 types of inputs | AI enabled Surveillance System |
| 3 | A text-based classifier which gives more scope to express the symptoms faced by the person based on the situations. | AI enabled Analysis |
| 4 | A basic UI to abstract the coding environment while in usage of the system. |  |

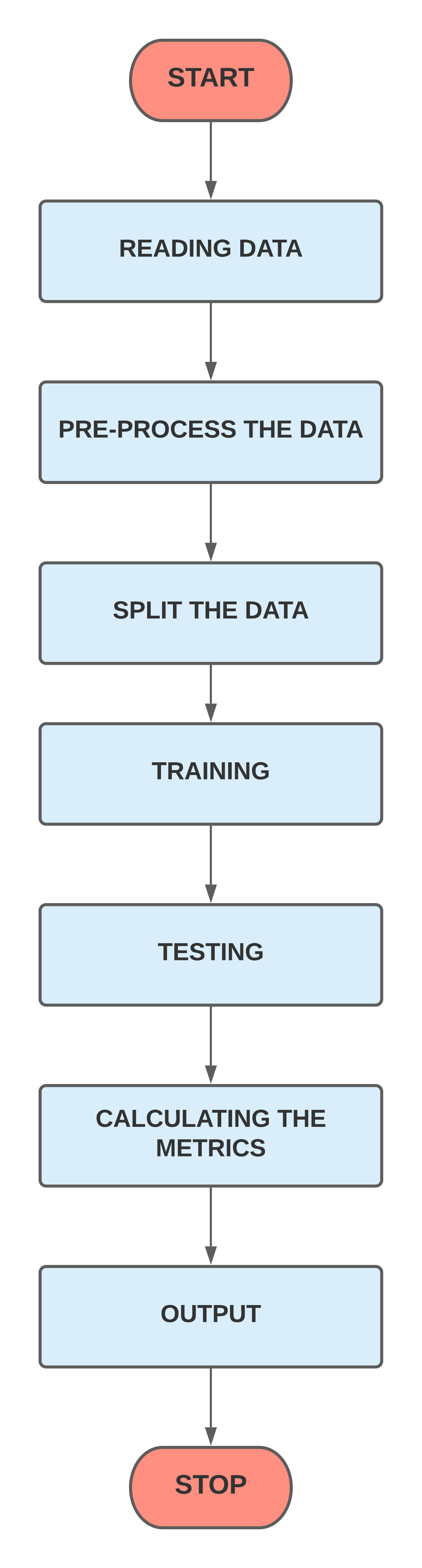
**CHAPTER 4 - SYSTEM DESIGN**

**4.1 UML**

**4.1.1 Data diagrams**

******

**4.1.2 Flow Chart**

******

**4.1.3 Flow chart**

A screenshot of a phone

Description automatically generated with low confidence

**4.1.4 Feasibility**

Diagram

Description automatically generated

**4.1.5 Class diagram**

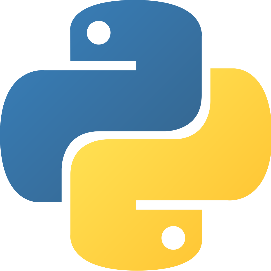
Diagram

Description automatically generated

**CHAPTER-5 SOFTWARE ENVIRONMENTS**

**5.1 Technologies used**

**5.1.1 Python**



Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

A screenshot of a computer

Description automatically generated

**5.1.2 OpenCV**

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human.

OpenCV is used for all sorts of image and video analysis, like facial recognition and detection, license plate reading, photo editing, advanced robotic vision, optical character recognition, and a whole lot more. We will be working through many Python examples here.

A screenshot of a computer

Description automatically generated

**5.1.3 Tkinter**

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit and is Python's de facto standard GUI. Tkinter is included with standard GNU/Linux, Microsoft Windows and macOS installs of Python. The name Tkinter comes from Tk interface.

It is used to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions. In fact, it's the only framework built into the Python standard library.

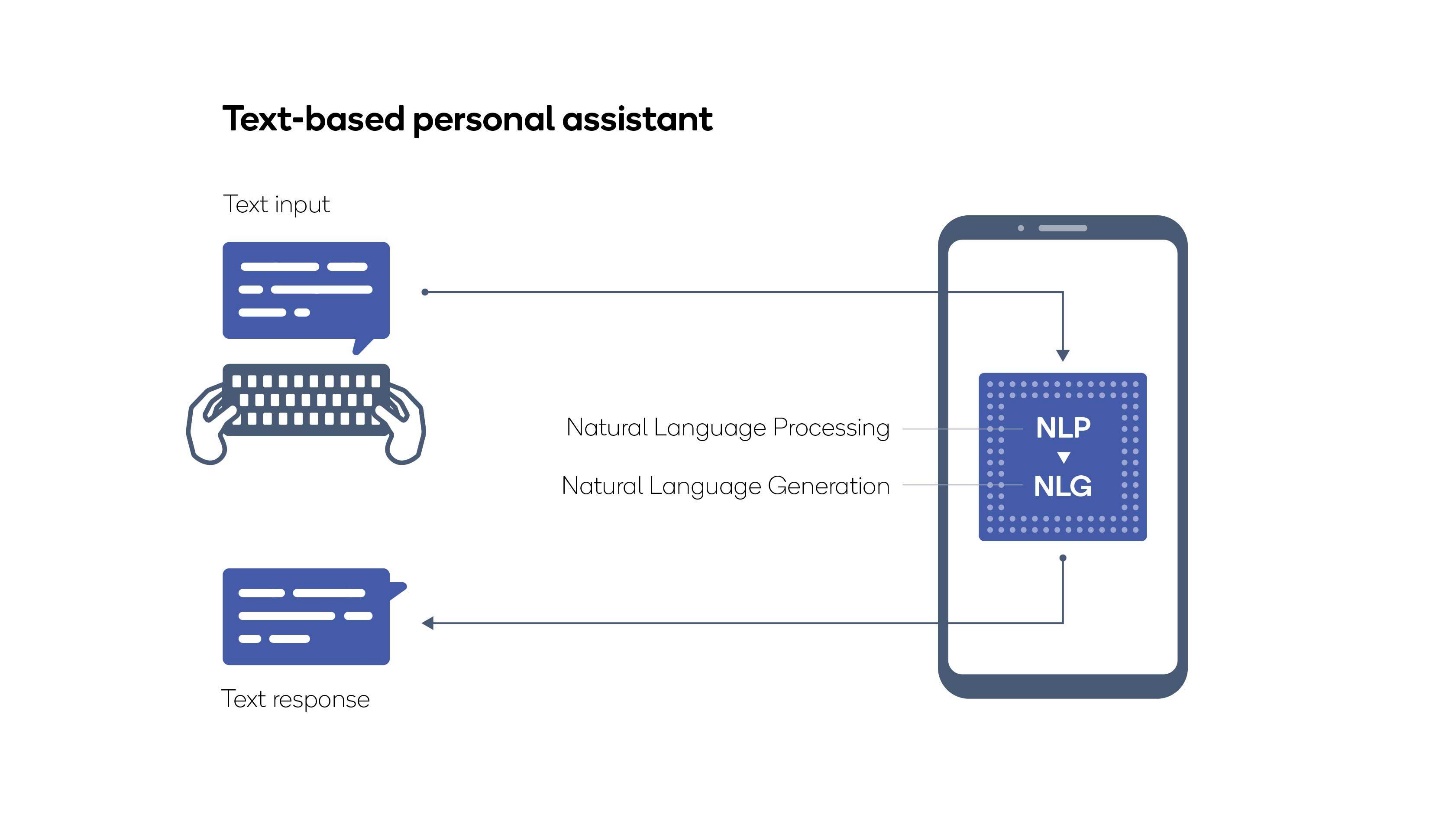
A screenshot of a computer

Description automatically generated

**5.2 Natural language processing** **(NLP)**

Natural language processing (NLP) refers to the branch of computer science and more specifically, the branch of [artificial intelligence](https://www.ibm.com/cloud/learn/what-is-artificial-intelligence) concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

NLP combines computational linguistic rule-based modeling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to ‘understand’ its full meaning, complete with the speaker or writer’s intent and sentiment.



**5.3 Machine learning**

**Support Vector Machine (SVM)**

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems.

In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

It is effective in high dimensional spaces.

It is also effective in cases where number of dimensions is greater than the number of samples.

It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

**5.4 Tensor flow**

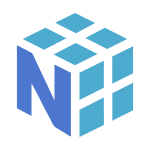


TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow is a symbolic math library based on dataflow and differentiable programming.

A screenshot of a computer

Description automatically generated

**5.5 NumPy**



NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

Graphical user interface, website

Description automatically generated

**Installation**

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

**5.2 Tools**

**5.2.1 Jupyter Notebook**

The Jupyter Notebook

Logo, company name

Description automatically generated

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Graphical user interface, application, Word

Description automatically generated

**5.2.2 Visual Studio Code**

Visual Studio Code



Visual Studio Code is an Integrated Development Environment made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Graphical user interface, text, application

Description automatically generated

**5.2.3** **Anaconda**

Anaconda



Anaconda is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS.

Graphical user interface, website

Description automatically generated

**5.3 Software environments**

**1.** **Conda**



Conda is an open-source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.

A screenshot of a computer

Description automatically generated

**CHAPTER-6 IMPLEMENTATION**

**6.1 Surveillance System**

**6.1.1 Libraries and Required files**

#!/usr/bin/env

import cv2

import numpy as np

import math

import os

from tkinter import \*

import tkinter as tk

from tkinter import messagebox

from tkinter import filedialog

harcascades = cv2.CascadeClassifier(r'C:/Users/Saicharan/haarcascades/haarcascade\_fullbody.xml')

harcascades\_2 = cv2.CascadeClassifier(r'C:/Users/Saicharan/haarcascades/haarcascade\_frontalface\_alt.xml')

**6.1.2 User Interface and Buttons**

window = Tk()

window.title("AI Enabled Surveillance and Analysis")

window.geometry('500x500')

lbl = Label(window, text="")

lbl.grid(column=55, row=0)

def browseFiles():

surveillance(0)

def browseImage():

filename = filedialog.askopenfilename(initialdir = "/",

title = "Select an Image",

filetypes = (("Image files",".jpg",),("all files",".")))

surveillance\_photo(filename)

def browseVideo():

filename = filedialog.askopenfilename(initialdir = "/",

title = "Select a Video",

filetypes = (("Video Files",

".mp4"),

("all files",

".")))

surveillance(filename)

btn = Button(window,text='Image', command=browseImage)

btn.grid(column=25, row=45)

btn = Button(window,text='Video', command=browseVideo)

btn.grid(column=45, row=45)

btn = Button(window,text='Live Feed', command=browseFiles)

btn.grid(column=75, row=45)

window.mainloop()

**6.1.3 Surveillance for LIVE and Video input**

def surveillance(filename):

cap = cv2.VideoCapture(filename)

i=0

while (cap.isOpened()):

ret,frame = cap.read()

try:

gray = cv2.cvtColor(frame,cv2.COLOR\_BGR2RGB)

except:

print("Video Ended")

break

faces = harcascades.detectMultiScale(gray,1.5,2)

no\_of\_faces = len(faces)

if len(faces) == 0:

pass

else:

for x,y,w,h in faces:

face = cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),7)

if no\_of\_faces >= 2:

x1 = faces[no\_of\_faces-2][0]

y1 = faces[no\_of\_faces-2][1]

w1 = x1 + faces[no\_of\_faces-2][2]

h1 = y1 + faces[no\_of\_faces-2][3]

else:

pass

if no\_of\_faces == 1:

distance = round(3200/(w\*12),2)

cv2.putText(frame,"Distance is " + str(distance) + "feet",(160,50),cv2.FONT\_HERSHEY\_SIMPLEX,1,(255,0,0),2,cv2.LINE\_AA)

else:

distance = math.sqrt( (x-x1) \*2 + (y-y1)\*2 )

actual\_distance = round(3200/(distance\*12),2)

cv2.putText(frame,"Distance is " + str(actual\_distance) + " feet",(30,50),cv2.FONT\_HERSHEY\_SIMPLEX,1,(255,0,0),2,cv2.LINE\_AA)

if actual\_distance >2:

cv2.putText(frame,'VIOLATING SOCIAL DISTANCING',(60,100),cv2.FONT\_HERSHEY\_SIMPLEX,1,(0,0,255),2,cv2.LINE\_AA,bottomLeftOrigin =False)

cv2.imwrite('e:/Hackathon/MINI PROJECT/Violated/violate'+str(i)+'.jpg',frame)

i=i+1

elif actual\_distance <=2:

cv2.putText(frame,'MAINTAINING SOCIAL DISTANCING',(60,110),cv2.FONT\_HERSHEY\_SIMPLEX,1,(0,255,0),2,cv2.LINE\_AA,bottomLeftOrigin =False)

try:

cv2.imshow('faces',frame)

except:

print(" ")

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

**6.1.4 Surveillance for Image input**

def surveillance\_photo(filename):

i =0

frame = cv2.imread(filename)

faces = harcascades\_2.detectMultiScale(frame,1.5,2)

no\_of\_faces = len(faces)

print(no\_of\_faces)

if len(faces) == 0:

pass

else:

for x,y,w,h in faces:

face = cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),7)

if no\_of\_faces >= 2:

x1 = faces[no\_of\_faces-2][0]

y1 = faces[no\_of\_faces-2][1]

w1 = x1 + faces[no\_of\_faces-2][2]

h1 = y1 + faces[no\_of\_faces-2][3]

else:

pass

if no\_of\_faces == 1:

distance = round(3200/(w\*12),2)

cv2.putText(frame,"Distance is " + str(distance) + "feet",(160,50),cv2.FONT\_HERSHEY\_SIMPLEX,1,(255,0,0),2,cv2.LINE\_AA)

else:

distance = math.sqrt( (x-x1) \*2 + (y-y1)\*2 )

actual\_distance = round(3200/(distance\*12),2)

cv2.putText(frame,"Distance is " + str(actual\_distance) + " feet",(30,50),cv2.FONT\_HERSHEY\_SIMPLEX,1,(255,0,0),2,cv2.LINE\_AA)

if actual\_distance <2:

cv2.putText(frame,'VIOLATING SOCIAL DISTANCING',(60,100),cv2.FONT\_HERSHEY\_SIMPLEX,1,(0,0,255),2,cv2.LINE\_AA,bottomLeftOrigin =False)

cv2.imwrite('e:/Hackathon/MINI PROJECT/Violated/violate'+str(i)+'.jpg',frame)

i=i+1

elif actual\_distance >=2:

cv2.putText(frame,'MAINTAINING SOCIAL DISTANCING',(60,110),cv2.FONT\_HERSHEY\_SIMPLEX,1,(0,255,0),2,cv2.LINE\_AA,bottomLeftOrigin =False)

**6.2 Analysis System**

**Library installed**

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.pipeline import Pipeline

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.svm import LinearSVC

from sklearn.metrics import confusion\_matrix

from sklearn import metrics

**Importing the data**

data = pd.read\_excel("e:/Hackathon/covid caases.xlsx")

**Removing null values**

blanks = []

for i,lab,rev in data.itertuples():

if rev.isspace():

blanks.append(i)

**Labelling the data**

x = data['Symptoms']

y = data['Label']

x.shape,y.shape

**Splitting the data**

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size = 0.3,random\_state = 117)

**Creating language pipeline**

clf\_mod = Pipeline([ ('tfidf',TfidfVectorizer()) , ('mod',LinearSVC()) ])

**Training the model**

clf\_mod.fit(x\_train,y\_train)

**Predicting values**

pred =clf\_mod.predict(x\_test)

**Accuracy**

print(confusion\_matrix(y\_test,pred))

print(f'Accuracy of the model is :{metrics.accuracy\_score(pred,y\_test)}')

print(f'{metrics.classification\_report(pred,y\_test)}')

**Output**

data2 = pd.DataFrame(question,columns = columns)

data2

print(clf\_mod.predict(data2['Customers Reviews']))

def Customer():

print("Enter Your Review")

columns = ['Customers Reviews']

string = input()

question = []

question.append(string)

data2 = pd.DataFrame(question,columns = columns)

print(clf\_mod.predict(data2['Customers Reviews']))

Customer()

Customer()

**CHAPTER-7 SCREENSHOTS**

**7.1 CMD Execution**

**Step 1**

Text

Description automatically generated

**Step 2**

Graphical user interface

Description automatically generated with low confidence

**Step 3**

Graphical user interface, website

Description automatically generated

**Step 4**

Graphical user interface

Description automatically generated

**OUTPUT**

A picture containing text, subway

Description automatically generated

**7.2 Live Monitoring**

**7.2.1**

Graphical user interface, website

Description automatically generated

**7.2.2**

A picture containing text, electronics, display

Description automatically generated

**7.2.3 Image**

A picture containing text, snow, people

Description automatically generated

**7.3 Prototype of Analysis software**

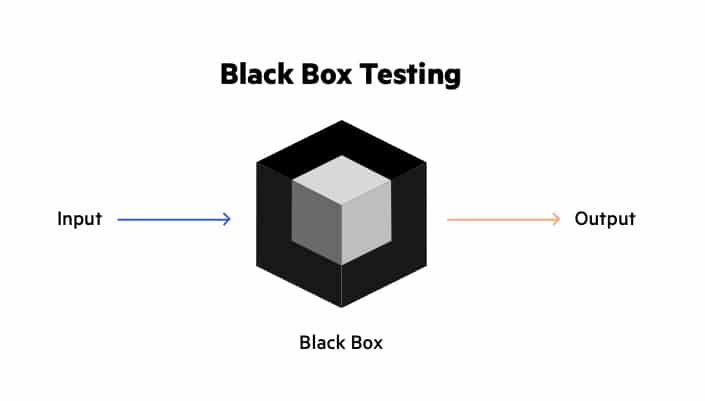
Graphical user interface, text, application

Description automatically generated

**CHAPTER-8 TESTING**

**8.1 Black Box Testing**

**Black Box Testing** is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

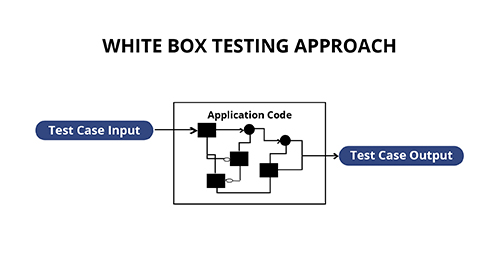


The above Black-Box can be any software system you want to test. For Example, an operating system like Windows, a website like Google, a database like Oracle or even your own custom application. Under Black Box Testing, you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

**8.2 White Box Testing**

**White Box Testing** is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers, so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing, and Glass box testing.

It is also called glass box testing or clear box testing or structural testing.



It is one of two parts of the Box Testing approach to software testing. Its counterpart, Blackbox testing, involves testing from an external or end-user type perspective. On the other hand, White box testing in software engineering is based on the inner workings of an application and revolves around internal testing.

The term “White Box” was used because of the see-through box concept. The clear box or White Box name symbolizes the ability to see through the software’s outer shell (or “box”) into its inner workings.

**CHAPTER-9 CONCLUSION**

Amidst the pandemic of the corona virus outbreak one of the strongest recommendations advised by all the governments in the world

Social distancing now is one of the greatest calls for humanity in these critical times.

In this way countries are not only using this is as a preventive measure for getting the prevention but also reducing the infection and flattening the curve of infection to community.

This project would help this cause of social distancing a lot by automatically monitoring social distancing and by reporting every instance whenever a violation of social distancing has occurred. It also helps in overcoming the burden of needing a human to monitor a crowd or a group of people and check whether there is a violation of social distancing happening.

This project would also help in overcoming the burden of needing to have to go through a whole video footage and checking whether there has been a violation of social distancing by recording and capturing only instances where social distancing has been violated.