MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

On

Motion Detection Using Open CV and Bokeh Plotting

Submitted By:

AdityaVardhan Singh Chohan

0901cs191009

Faculty Mentor:

Mr. Mir Shahnawaz Ahmad

Assistant Professor, Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Motion Detection Using Open CV and Bokeh Plotting

A project report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted By:

AdityaVardhan Singh Chohan

0901cs191009

Faculty Mentor:

Prof. Mir Shahnawaz Ahmad

Assistant Professor, Computer Science and Engineering

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957
MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **AdityaVardhan Singh Choahn** (0901cs1191009) has submitted the project report titled **motion detection using open cv and bokeh plotting** [illegible] under the mentorship of **Mr.Mir Shahnawaz Ahmad** in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

09/05/22

Mr. Mir Shahnawaz Ahmad
Faculty Mentor
Professor
Computer Science and Engineering

09/05/2022

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute
Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Prof.Mir Shahnawaz Ahmad** , **Professor** , **Computer science.**

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

AdityaVardhan Singh Chohan
0901CS191009
3rd Year
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to
RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this project. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Prof.Mir Shahnawaz Ahmad , Professor , Computer science ,**for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

AdityaVardhan Singh Chohan
0901CS191009
3rd Year,
Computer Science and Engineering

ABSTRACT

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. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

This OpenCV tutorial will help you learn the Image-processing from Basics to Advance, like operations on Images, Videos using a huge set of Opencv-programs and projects.

Also libraries like pandas and numpy help the same to create this project.The purpose of this paper is to introduce and quickly make a reader familiar with OpenCV (Open Source Computer Vision) basics without having to go through the lengthy reference manuals and books. OpenCV is an open source library for image and video analysis, originally introduced more than decade ago by Intel. Since then, a number of programmers have contributed to the most recent library developments.

सार:

OpenCV कंप्यूटर विज़न, मशीन लर्निंग और इमेज प्रोसेसिंग के लिए एक विशाल ओपन-सोर्स लाइब्रेरी है। ओपनसीवी विभिन्न प्रकार की प्रोग्रामिंग भाषाओं जैसे पायथन, सी ++, जावा, आदि का समर्थन करता है। यह वस्तुओं, चेहरों या यहां तक कि मानव की लिखावट की पहचान करने के लिए छवियों और वीडियो को संसाधित कर सकता है। जब इसे विभिन्न पुस्तकालयों के साथ एकीकृत किया जाता है, जैसे कि नम्पी जो संख्यात्मक संचालन के लिए एक अत्यधिक अनुकूलित पुस्तकालय है, तो आपके शस्त्रागार में हथियारों की संख्या बढ़ जाती है यानी नम्पी में कोई भी ऑपरेशन जो भी कर सकता है उसे ओपनसीवी के साथ जोड़ा जा सकता है।

यह ओपनसीवी ट्यूटोरियल आपको बेसिक से एडवांस तक इमेज-प्रोसेसिंग सीखने में मदद करेगा, जैसे ओपनसीवी-प्रोग्राम्स और प्रोजेक्ट्स के विशाल सेट का उपयोग करके इमेज, वीडियो पर ऑपरेशन।

साथ ही पांडा और numpy जैसे पुस्तकालय इस परियोजना को बनाने में मदद करते हैं। इस पेपर का उद्देश्य लंबे संदर्भ मैनुअल और पुस्तकों के माध्यम से जाने के बिना ओपनसीवी (ओपन सोर्स कंप्यूटर विजन) मूल बातें से परिचित कराना और जल्दी से पाठक को परिचित करना है। ओपनसीवी छवि और वीडियो विश्लेषण के लिए एक ओपन सोर्स लाइब्रेरी है, जिसे मूल रूप से इंटेल द्वारा एक दशक से अधिक समय पहले पेश किया गया था। तब से, कई प्रोग्रामर्स ने हाल के पुस्तकालय विकास में योगदान दिया है।

TABLE OF CONTENTS

TITLE

List of figures .

Chapter 1: Project Overview

In this project we have used many kind of libraries to excute the give problem state ment which are liste as under

1.1 **OpenCV-Python** is a library of Python bindings designed to solve computer vision problems

1.2 **PANDAS** is a Python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the **fundamental high-level building block for doing practical, real-world data analysis in Python**.

1.3 **NumPy** is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices

1.4 **BOKEH** is a data visualization library in Python that provides high-performance interactive charts and plots. Bokeh output can be obtained in various mediums like notebook, html and server. It is possible to embed bokeh plots in Django and flask apps.

Chapter 2 – Literature Overview

2.1)In order to capture video we will be using while loop.

While condition will be such that unless check is true, Python willdisplay the frames.

Firstly, we will be creating a video capture object, then within the while loop the first frame will be captured which is actually a numpy array object.We will be converting this frame to grayscale image by using (cvtColor) library in OpenCV.

After that we will be generating a new frame after each iteation in while loop.

2.2) Problem Statement

2.3)So the situation could be that we have been approached by a company that is studying human behavior. Your task is to give them a webcam, that can detect the motion or any movement in front of it. This should return a graph, this graph should contain for how long the human/object was in front of the camera

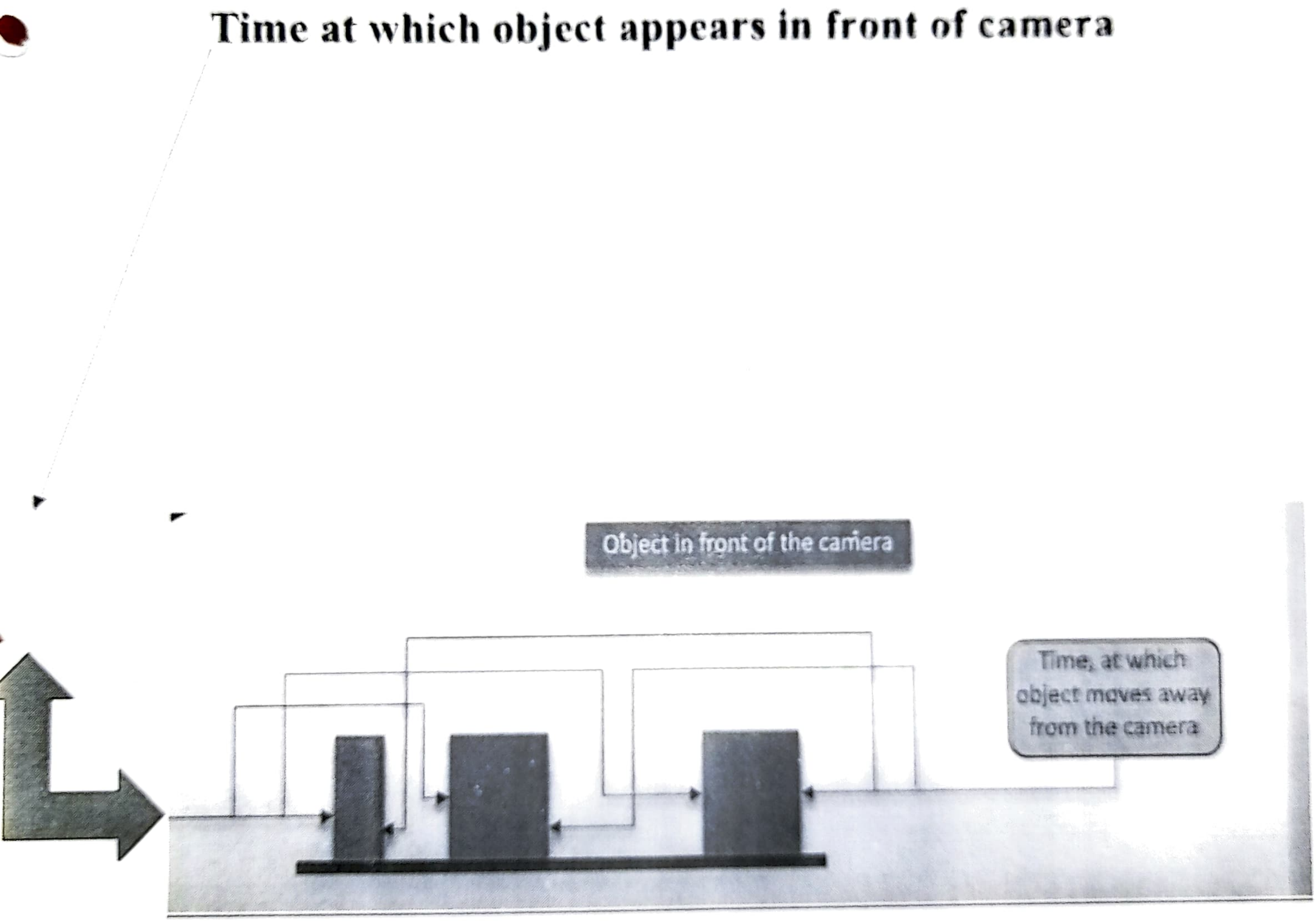

Time at which object appears in front of camera
Object in front of the camera
Time, at which object moves away from the camera

Figure 1.0 Problem Statement

2.4) Solution logic

4.1) A SOLUTION logic model is a schematic representation that describes how a program* is intended to work by linking activities with outputs, intermediate impacts and longer term outcomes. Program logic aims to show the intended causal links for a program

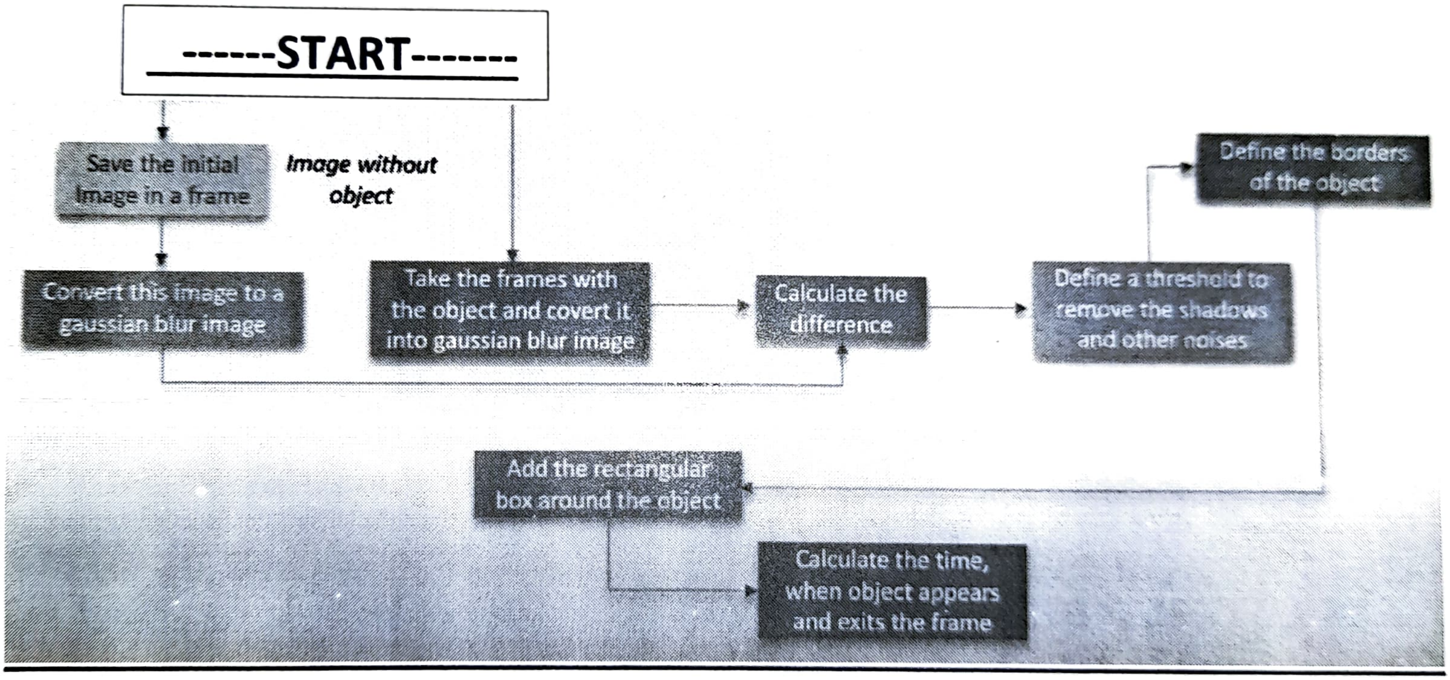

------START-------
Save the initial Image in a frame
Image without object
Convert this image to a gaussian blur image
Take the frames with the object and covert it into gaussian blur image
Calculate the difference
Define a threshold to remove the shadows and other noises
Define the borders of the object
Add the rectangular box around the object
Calculate the time, when object appears and exits the frame

Figure 1.1 Solution logic

Chapter-3 Preliminary Design

3.1) Storing first image/frame of video.

3.2)Calculating difference between first and other frames.

3.3)Providing threshold value that with value less than 30 will be black and white if greater than it

3.4)Adding borders to object that appears

3.5)Removing unwanted small object, will keep those having area greater than 1000px.

3.6)Changing status when object is being detected

3.7)Creating rectangular box

3.8)Record datetime in list when object occurs //Storing time values

3.9)Showing 4 different frames

3.10)Changing frame after 1 millisecond

3.11)To close the window when Q pressed

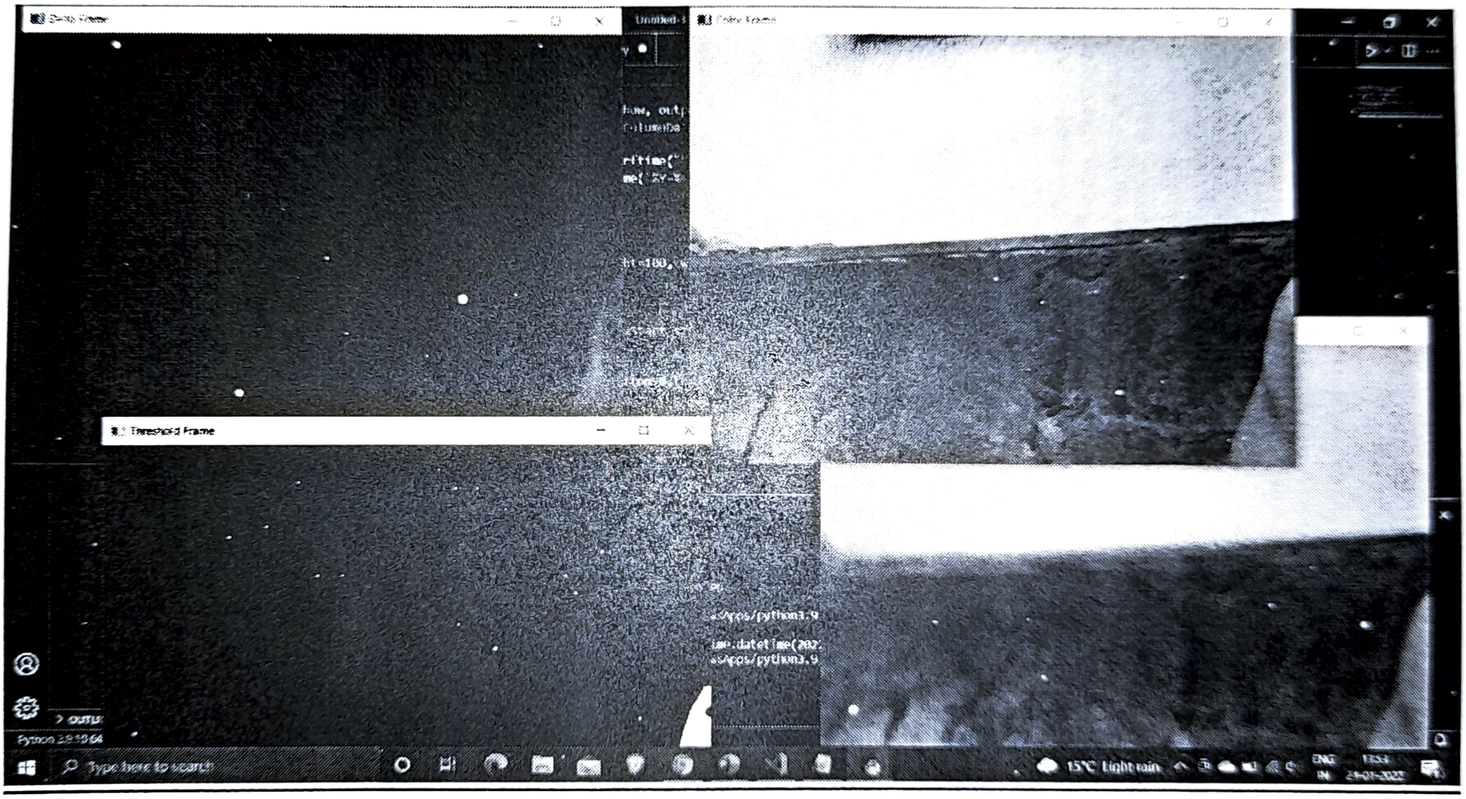
3.12)Storing time values in data frame

3.13)Writing to csv file

4.0)Code in action-Before any unidentified object

4.1) Representation Of code running. When The Object is Not

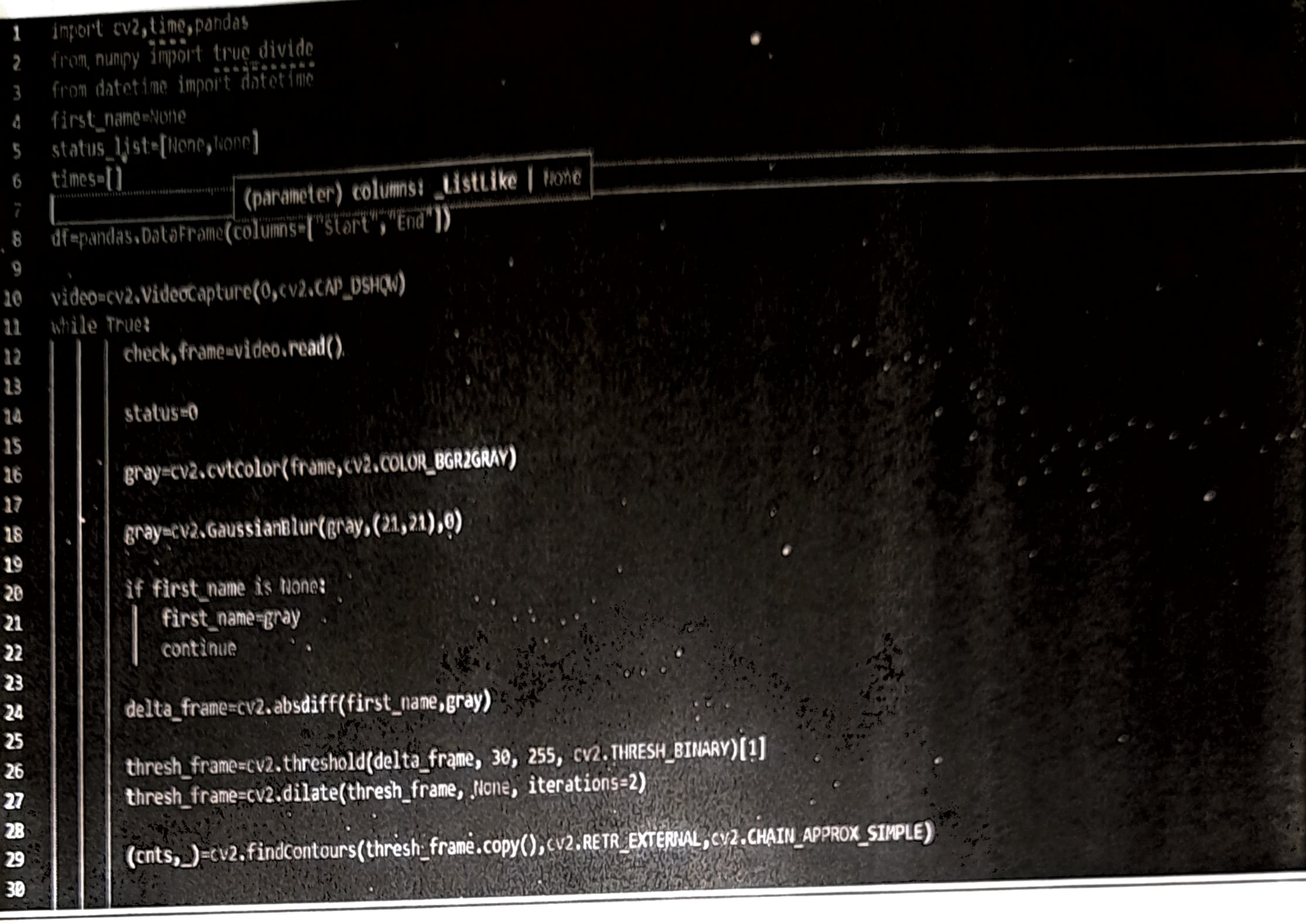
Present in frontT of.. the Screen we can see in the given figure 1.2 that we can see there is no object till now.



```
hme, outp
C:\UsersDa
rftime("
me( %Y-%
ht=180,
sApps/python3.9
ime.datetime(202
sApps/python3.9
```

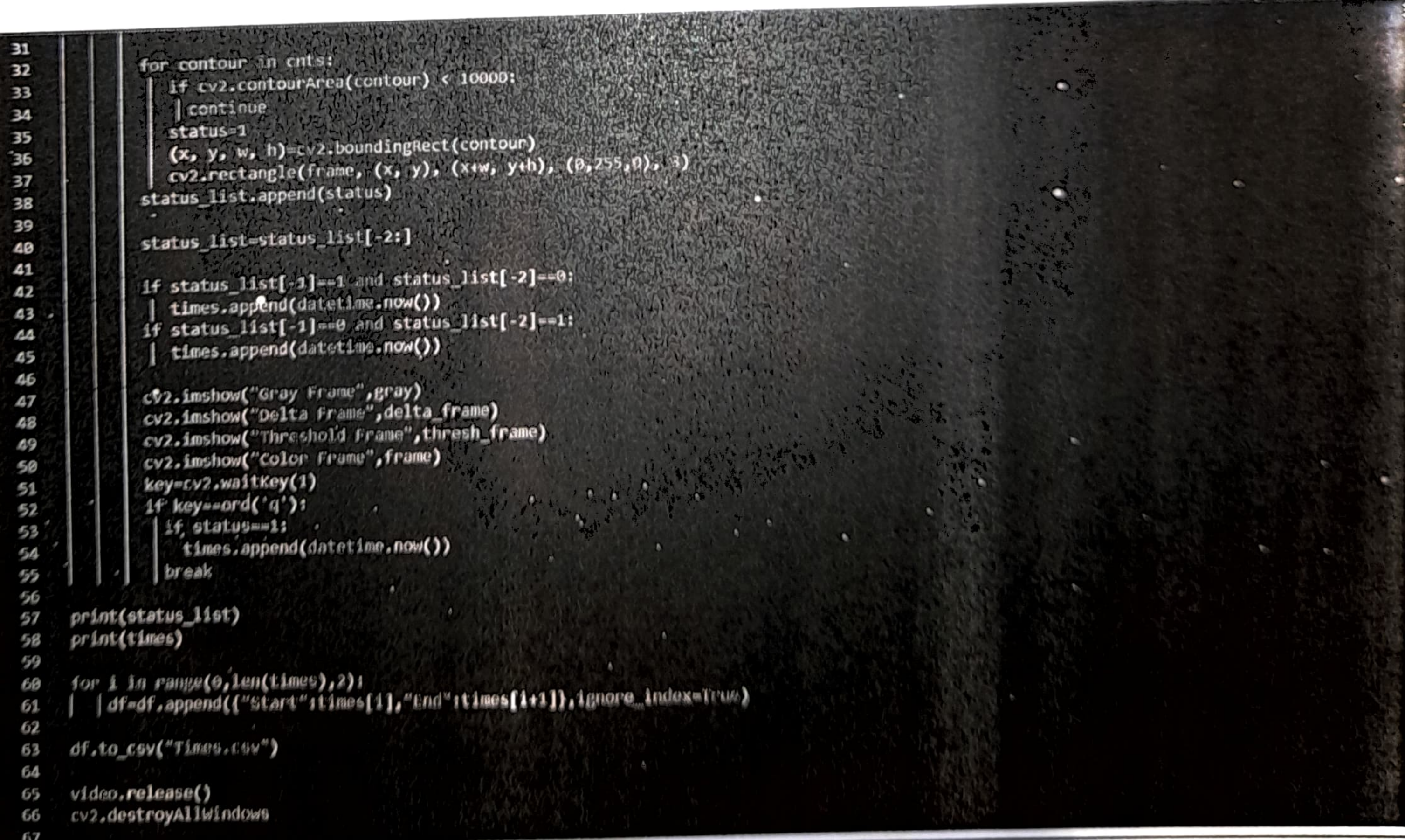
Figure 1.2 object without screen

4.5 - Motio



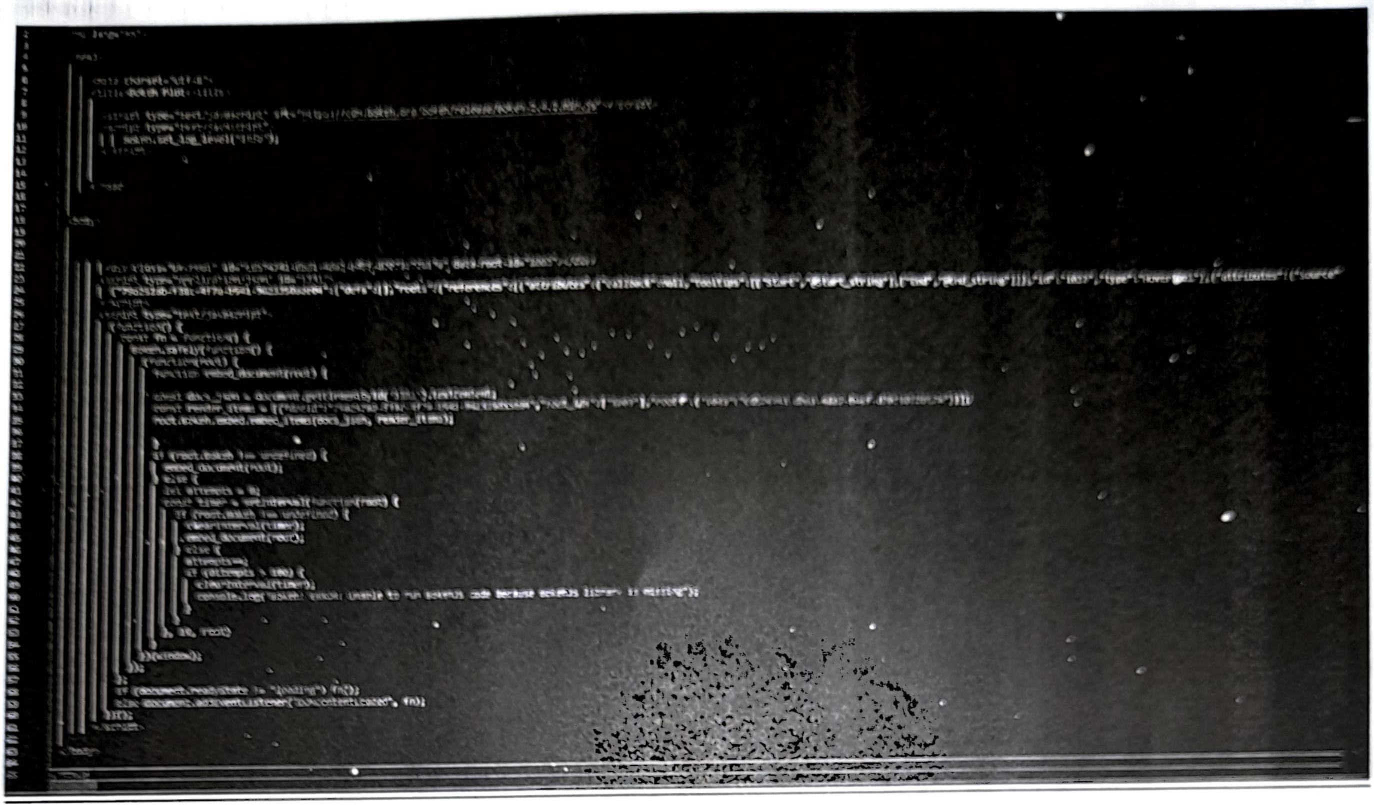
```python
import cv2,time,pandas
from numpy import true_divide
from datetime import datetime
first_name=None
status_list=[None,None]
times=[]

df=pandas.DataFrame(columns=["Start","End"])

video=cv2.VideoCapture(0,cv2.CAP_DSHOW)
while True:
    check,frame=video.read()

    status=0

    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)

    gray=cv2.GaussianBlur(gray,(21,21),0)

    if first_name is None:
        first_name=gray
        continue

    delta_frame=cv2.absdiff(first_name,gray)

    thresh_frame=cv2.threshold(delta_frame, 30, 255, cv2.THRESH_BINARY)[1]
    thresh_frame=cv2.dilate(thresh_frame, None, iterations=2)

    (cnts,_)=cv2.findContours(thresh_frame.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)

```

Figure 1.3



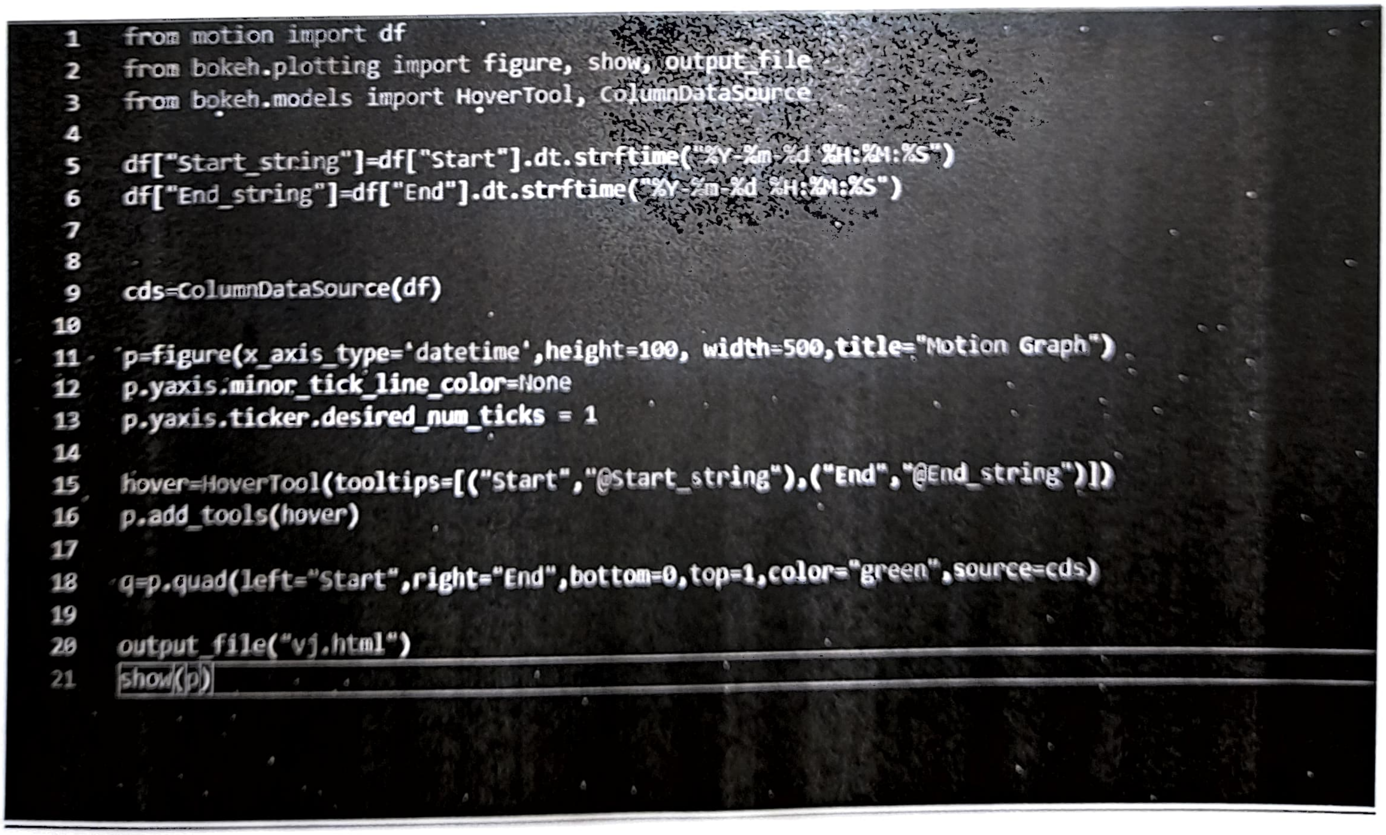
```python

    for contour in cnts:
        if cv2.contourArea(contour) < 10000:
            continue
        status=1
        (x, y, w, h)=cv2.boundingRect(contour)
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0,255,0), 3)
    status_list.append(status)

    status_list=status_list[-2:]

    if status_list[-1]==1 and status_list[-2]==0:
        times.append(datetime.now())
    if status_list[-1]==0 and status_list[-2]==1:
        times.append(datetime.now())

    cv2.imshow("Gray Frame",gray)
    cv2.imshow("Delta Frame",delta_frame)
    cv2.imshow("Threshold Frame",thresh_frame)
    cv2.imshow("Color Frame",frame)
    key=cv2.waitKey(1)
    if key==ord('q'):
        if status==1:
            times.append(datetime.now())
        break

print(status_list)
print(times)

for i in range(0,len(times),2):
    df=df.append({"Start":times[i],"End":times[i+1]},ignore_index=True)

df.to_csv("Times.csv")

video.release()
cv2.destroyAllWindows

```

4.4 Plotting the motion detection graph

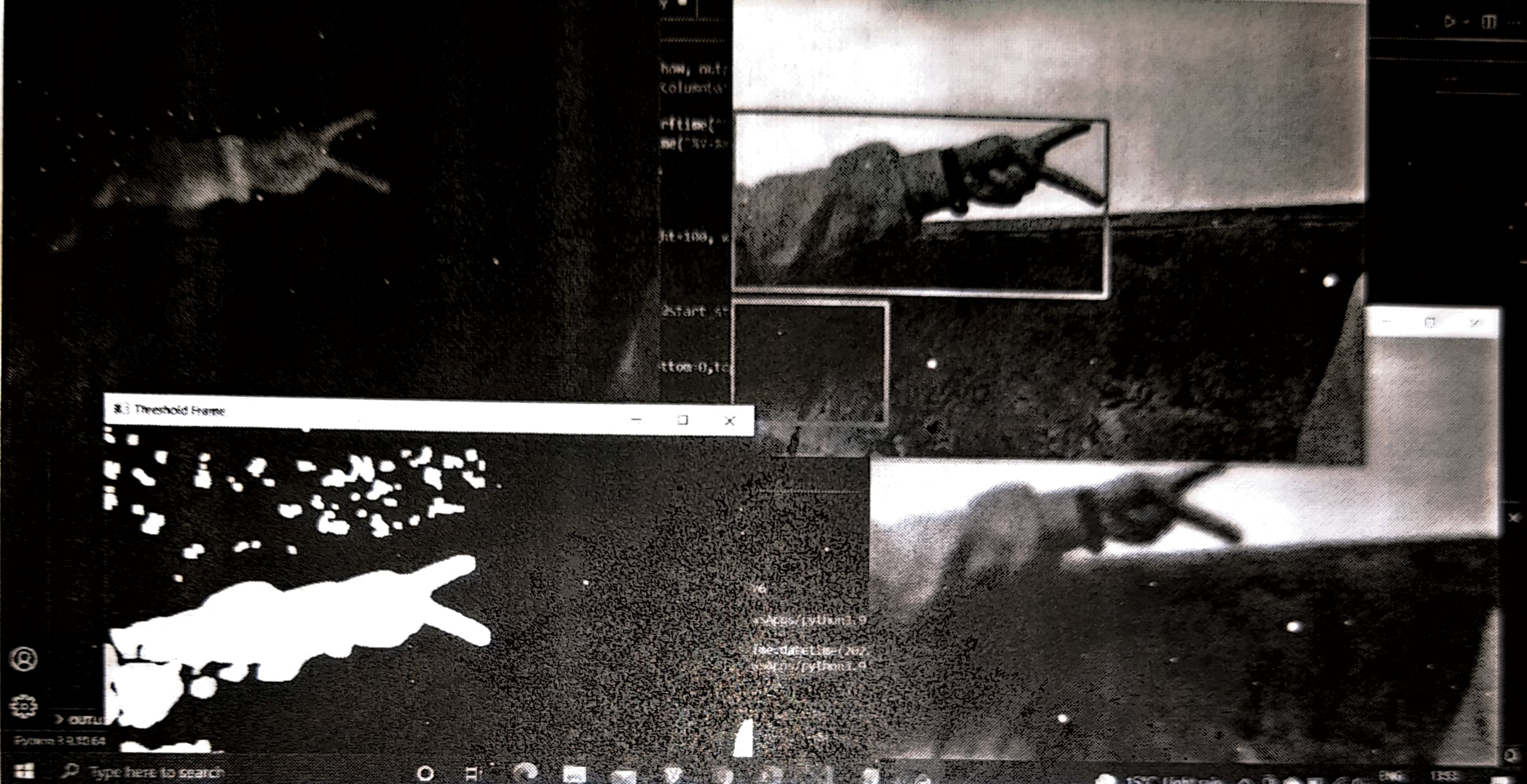


```
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Bokeh Plot</title>
    <script type="text/javascript" src="https://cdn.bokeh.org/bokeh/release/bokeh-[illegible].min.js"></script>
    <script type="text/javascript">
        Bokeh.set_log_level("info");
    </script>
  </head>
  <body>
    <div class="bk-root" id="[illegible]" data-root-id="[illegible]"></div>
    <script type="application/json" id="[illegible]">
      {"[illegible]":{"defs":[],"roots":{"references":[{"attributes":{"callback":null,"tooltips":[["Start","@Start_string"],["End","@End_string"]]},"id":"[illegible]","type":"HoverTool"},{"attributes":{"source":[illegible]
    </script>
    <script type="text/javascript">
      (function() {
        const fn = function() {
          Bokeh.safely(function() {
            (function(root) {
              function embed_document(root) {

              const docs_json = document.getElementById('[illegible]').textContent;
              const render_items = [{"docid":"[illegible]","root_ids":["[illegible]"],"roots":{"[illegible]":"[illegible]"}}];
              root.Bokeh.embed.embed_items(docs_json, render_items);

              }
              if (root.Bokeh !== undefined) {
                embed_document(root);
              } else {
                let attempts = 0;
                const timer = setInterval(function(root) {
                  if (root.Bokeh !== undefined) {
                    clearInterval(timer);
                    embed_document(root);
                  } else {
                    attempts++;
                    if (attempts > 100) {
                      clearInterval(timer);
                      console.log("Bokeh: ERROR: Unable to run BokehJS code because BokehJS library is missing");
                    }
                  }
                }, 10, root)
              }
            })(window);
          });
        };
        if (document.readyState != "loading") fn();
        else document.addEventListener("DOMContentLoaded", fn);
      })();
    </script>
  </body>
</html>
```

Figure 2.0,2.1



```
from motion import df
from bokeh.plotting import figure, show, output_file
from bokeh.models import HoverTool, ColumnDataSource

df["Start_string"]=df["Start"].dt.strftime("%Y-%m-%d %H:%M:%S")
df["End_string"]=df["End"].dt.strftime("%Y-%m-%d %H:%M:%S")


cds=ColumnDataSource(df)

p=figure(x_axis_type='datetime',height=100, width=500,title="Motion Graph")
p.yaxis.minor_tick_line_color=None
p.yaxis.ticker.desired_num_ticks = 1

hover=HoverTool(tooltips=[("Start","@Start_string"),("End","@End_string")])
p.add_tools(hover)

q=p.quad(left="Start",right="End",bottom=0,top=1,color="green",source=cds)

output_file(“vj.html“)
show(p)
```


Threshold Frame
how, out
columns
rftime("
me("%Y-%
ht+100, w
start
ttom:0,tc
Apps/python3.9
Time:datetime(202
Apps/python3.9
OUTLINE
Python 3.9.10 64
Type here to search

4.5)Graph plotting in action

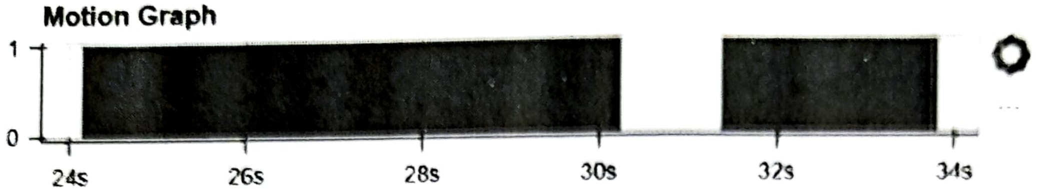
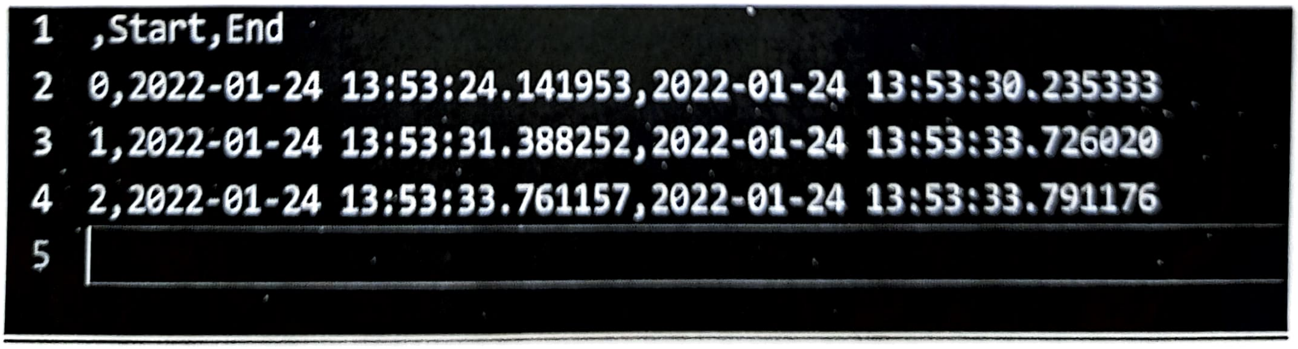


Figure 2.3 object time graph

10.1 The above graph shows the timing of the object when it crosses the screen.

TIME CSV FILE –



```
,Start,End
0,2022-01-24 13:53:24.141953,2022-01-24 13:53:30.235333
1,2022-01-24 13:53:31.388252,2022-01-24 13:53:33.726020
2,2022-01-24 13:53:33.761157,2022-01-24 13:53:33.791176
```

Figure 2.4 time table of objects enterdd the screen with date.

Chapter 5 Future Scope And Conclusion.

5.1) Future Scope:

The future Scope of my project is thet it can be used as a proctored exams and face recognition devices etc although it can successfully detect and object passing the screen during a particular time period and the we get our proper record of objects corssinf the screen further more advancement can lead to the face recognition protoype by using certain libraries

5.2) Conclusion:

Hence we can conclude by saying that Open Cv using bokeh plotting can be used as a object detection techniuque during certai conditons.

Refrences

1. Learning open CV and computer vision by - Adrian Kaehler, Gary Rost Bradski
2. Edureak Open Cv crash course

3.Nptel The Joy of Computoing With Python.