A MINI PROJECT REPORT ON

SMART CAMERA

A dissertation submitted in partial fulfilment of the

Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

Submitted by

T.Arya (19B81A12C2)

Ch.Mukesh (19B81A12E2)

S.Pratheek (19B81A12H1)

Under the esteemed guidance of

Mrs.Swathi Agarwal

Assistant Professor, IT Department

CVR College of Engineering

DEPPARTMENT OF INFORMATION TECHNOLOGY

CVR COLLEGE OF ENGINEERING

ACCREDITED BY NBA, AICTE & Affiliated to JNTU-H Vastunagar,

Mangalpally (V), Ibrahimpatnam (M), R.R. District, PIN-501 510 2022-2023

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the Project Report entitled "SMART CAMERA "is a bonafide work done and submitted by **T.Arya(19B81A12C2), Ch.Mukesh(19B81A12E2), S.Pratheek(19B81A12H1**) during the academic year 2022-2023, in partial fulfilment of requirement for the award of Bachelor of Technology degree in Information Technology from Jawaharlal Nehru Technological University Hyderabad, is a bonafide record of work carried out by them under my guidance and supervision**.**

Certified further that to my best of the knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

INTERNAL GUIDE	**HEAD OF THE DEPARTMENT**
Mrs. Swathi Agarwal	**Dr.Bipin Bihari Jayasingh**
Assistant Professor, IT Department	Professor, IT Department
PROFESSOR INCHARGE	**PROJECT COORDINATOR**
Mr.C.V.S Satyamurty	**Mrs.G. SunithaRekha**
Professor, IT Department	Associate Professor, IT Department

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction of completing this project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

First and foremost, we thank the Chairman, Principal, Vice Principal for availing infrastructural facilities to complete the mini project in time.

We offer our sincere gratitude to our internal guide **Mrs. Swathi Agarwal**, Assistant Professor, IT Department, CVR College of Engineering for his immense support, timely co-operation and valuable advice throughout the course of our project work.

We would like to thank the Professor In-Charge of Projects**, Mr. C.V.S Satyamurty**, Professor, Information Technology for his valuable suggestions in implementing the project.

We would like to thank the Head of Department, Professor **Dr. Bipin Bihari Jayasingh**, for his meticulous care and cooperation throughout the project work.

We are thankful to **Mrs. G. Sunitha Rekha**, Project Coordinator, Assistant Professor, IT Department, CVR College of Engineering for his supportive guidelines and for having provided the necessary help for carrying forward this project without any obstacles and hindrances.

We also thank the **Project Review Committee Members** for their valuable suggestions.

DECLARATION

We hereby declare that the project report entitled “SMART CAMERA” is an original work done and submitted to IT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfilment of the requirement for the award of Bachelor of Technology in **Information Technology** and it is a record of bonafide project work carried out by us under the guidance of **Mrs. Swathi Agarwal, Assistant Professor, Department of Information Technology.**

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

T.ARYA

(19B81A12C2)

CH. MUKESH

(19B81A12E2)

S.PRATHEEK

(19B81A12H1)

ABSTRACT

Security is a crucial concern now-a-days and there are lot of technologies present today to keep your place secure and monitored but these technologies cost more for domestic and small-scale business where security is required but with less expenditure and maintenance. Upon that, in traditional CCTV cameras, it is required for a person to continuously monitor the system. The main aim of this project is creating an automated camera which can detect, monitor and alert the user without any human intervention. Our project is focused on automating the surveillance process with the help of camera using an Python Libraries and IQA algorithm. Compared to traditional camera system. Smart camera detects a person who is in the camera vision. When the person is detected, it starts recording, motion detection, motion capture, anti-theft, face identification and these recordings are stored in local database. It is helpful in place of CCTV and also for personal use. It basically does motion detection. We can increase the surveillance system at a very low cost and easy implementation.

LIST OF FIGURES

LISTS OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Camera is bascially a device that consists of a lightproof chamber with an aperture fitted with a lens and a shutter through which the image of an object is projected onto a surface for recording (as on a photosensitive film or an electronic sensor) or for translation into electrical impulses (as for television broadcast). There are many different types of cameras such as Compact Digital/ Point and Shoot cameras , Bridge cameras , Action cameras , 360 cameras , Smartphone cameras ,Film cameras, Rugged cameras and Security or Surveilence cameras.

Surveillance of the public using CCTV is common in many areas around the world. The deployment of this technology has facilitated significant growth in state surveillance, a substantial rise in the methods of advanced social monitoring and control, and a host of crime prevention measures throughout the world. In industrial plants, CCTV equipment may be used to observe parts of a process from a central control room, especially if the environments observed are dangerous or inaccessible to humans. CCTV systems may operate continuously or only as required to monitor a particular event. A more advanced form of CCTV, using digital video recorders (DVRs), provides recording for possibly many years, with a variety of quality and performance options and extra features (such as motion detection and email alerts).

Surveillance cameras or security cameras, are video cameras used for the purpose of observing an area. They are often connected to a recording device or IP network, and may be watched by a security guard or law enforcement officer. Cameras and recording equipment used to be relatively expensive and required human personnel to monitor camera footage, but analysis of footage has been made easier by automated software that organizes digital video footage into a searchable database, and by video analysis software.

Smart Camera does various functions and works even without any human intervention. This smart camera has different features which can perform by using monitoring, identifying, motion detection , visitors capture and recording .This project is made using Python Libraries and IQA algorithm.

LITERATURE REVIEW

Referred Papers	Name Of the author(s)	Extracted Topics
Anti-theft monitoring using IQA Algorithms[1] **https://www.researchgate.net/publication/342435717_Understanding_SSIM**	Sheikh, H. R., & Bovik, A. C	SSIM algorithm
Evaluation of Haar Cascade Classifiers for Face Detection[2] **https://www.researchgate.net/publication/303251696**	Rafael Padilla,Marly Costa.	Evaluation of Haar Cascade Classifiers
OpenCV for Computer Vision Applications[3] **https://www.researchgate.net/publication/301590571**	Naveenkumar, Vadivel	Working of Open Cv, Image processing.
Face Identification Using machine learning[4] **https://www.ijcrt.org/papers/IJCRT2005173.pdf**	B.Kameswara Rao, Anusha Baratam,	LBPH
Motion Detection Algorithms[5] **https://www.irjet.net/archives/V9/i2/IRJET-V9I2170.pdf**	Pannaga Siri, Dr.Ganesh	Motion Detection techniques

Table 1: Literature Review

[1] The Structural Similarity Index (SSIM) is a perceptual metric that quantifies image quality degradation caused by processing such as data compression or by losses in data transmission. It is a full reference metric that requires two images from the same image capture— a reference image and a processed image. The processed image is typically compressed. It may, for example, be obtained by saving a reference image as a JPEG (at any quality level) then reading it back in. SSIM is best known in the video industry, but has strong applications for still photography. SSIM is used as a metric to measure the similarity between two given images.A lot of material exists explaining the theory behind SSIM but very few resources go deep into the details, that too specifically for a gradient-based implementation as SSIM is often used as a loss function.

[2] Vision-based automated systems applied to face recognition can be divided into 4 steps: face detection, image pre-processing. Face detection is a hard task, once faces form a similar class of objects and their features, such as eyes, mouth, nose and chin, have, in general, the same geometrical configuration. Implementations of this framework, such as OpenCV, provide different face classifiers created by authors that used different datasets into their training. The performance and reliability of these classifiers vary a lot.In this paper, we learned more abothaar cascade classifiers.

[3] The aim of image processing is to help the computer to understand the content of an image. OpenCV is a library of programming functions mainly used for image processing.We can solve many real time problems using image processing applications. In this paper, sample real time image processing applications of OpenCV are discussed along with steps.OpenCV contains various tools to solve computer vision problems. It contains low level image processing functions and high level algorithms for face detection, feature matching and tracking.

[4] Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. Due to its discriminative power and computational simplicity, LBP texture operator has become a popular approach in various applications. It can be seen as a unifying approach to the traditionally divergent statistical and structural models of texture analysis. Perhaps the most important property of the LBP operator in real-world applications is its robustness to monotonic gray-scale changes caused, for example, by illumination variations. Another important property is its computational simplicity, which makes it possible to analyze images in challenging real-time settings.

[5] In this paper, sample real time image processing applications of OpenCV are discussed along with steps.OpenCV contains various tools to solve computer vision problems. It contains low level image processing functions and high level algorithms for face detection, feature matching and tracking.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATIONS

2.1 FUNCTIONAL REQUIREMENTS:

Following are various functional requirements of the project:

- The system must be able to identify human faces in live video.
- It must be able to detect the face using classifiers and then show the results.
- The system must accommodate a hd camera.
- The system must be able to extract each face’s Region of Interest (ROI).

2.2 NON - FUNCTIONAL REQUIREMENTS:

Following are various non - functional requirements of the project:

- The face should be localized by detecting the facial landmarks and the background must be ignored.
- The system will be implemented in Python script with an accuracy of the model of over 75%.
- The user must not move his/her face out of camera’s sight in order to get correct results.
- The background must not be too bright or too dark while detecting.
- The system should be easy for usability and self-descriptive for maintenance purpose.

2.3 HARDWARE AND SOFTWARE REQUIREMENTS:

HARDWARE REQUIREMENTS:

• Monitor	: Camera integrated system
• Hard-disk	: 240 GB ssd
• RAM	:8GB
• Keyboard , mouse	

SOFTWARE REQUIREMENTS:

• Operating System	: Windows OS, Linux Mint, Linux Ubunutu
• Language	: Python
• IDE	: Visual Studio Code
• Libraries	: openCV, Pillow, Tkinter, Skimage, numpy
• Technologies	: Machine Learning

SOFTWARE ARCHITECTURE:

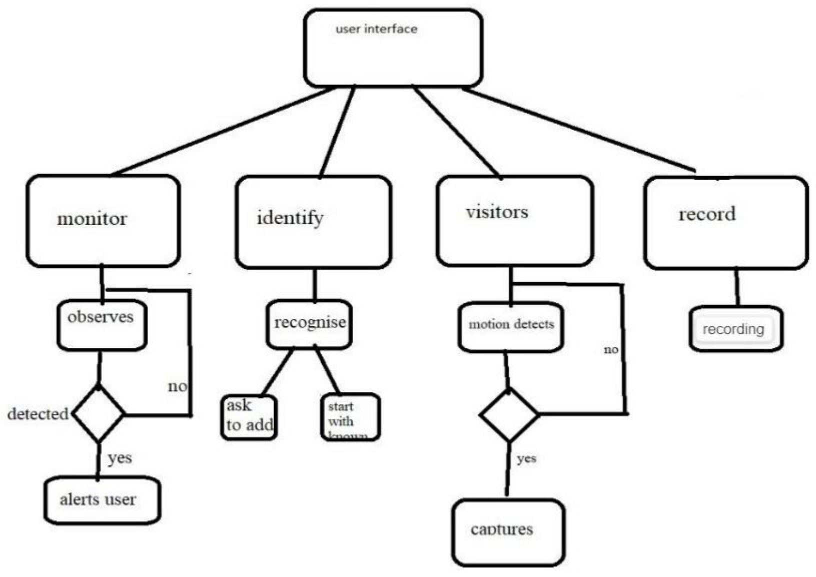


Fig 1 : Software Architecture Diagram

CHAPTER 3

DESIGN

USE CASE DIAGRAM :

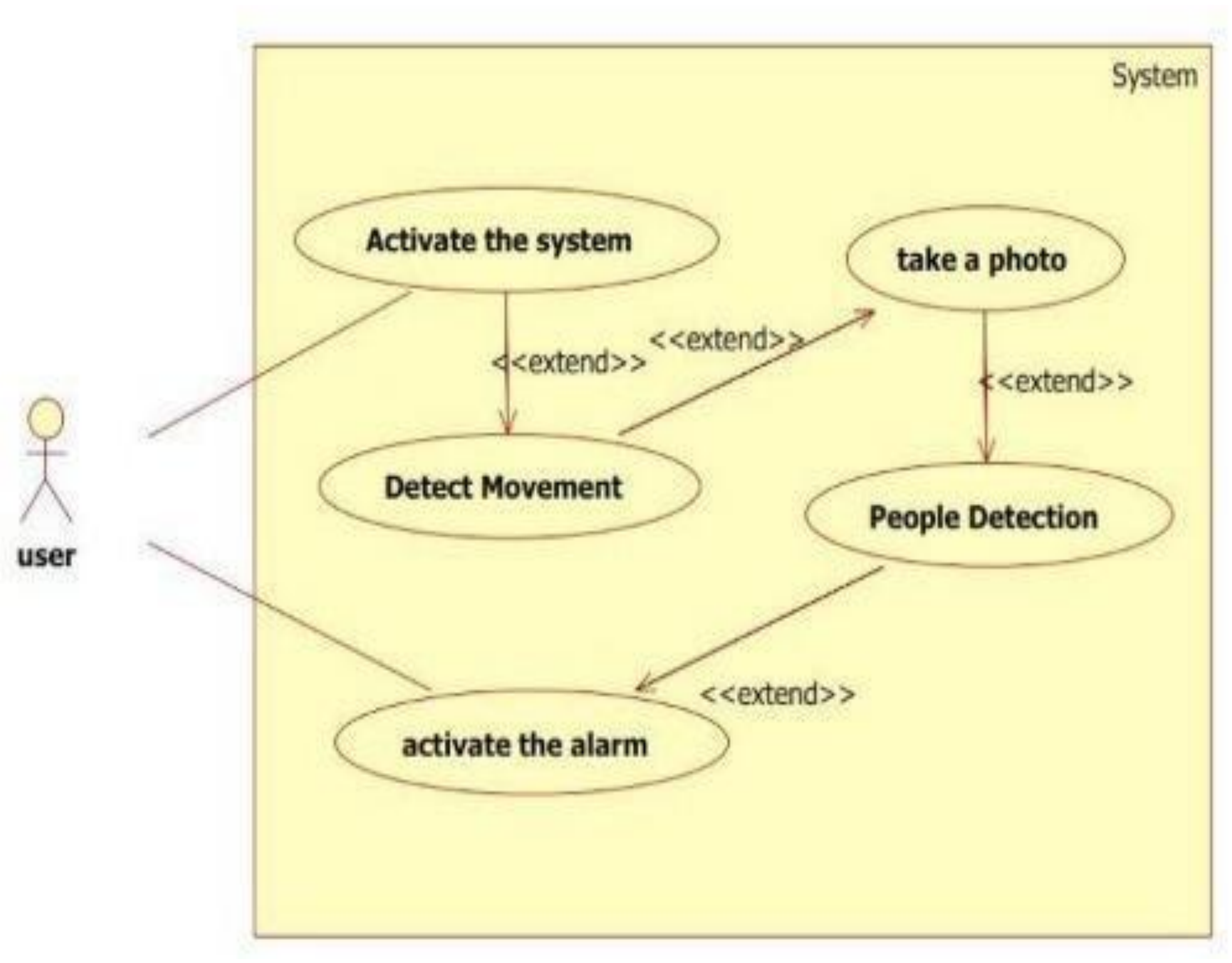


Fig 2: Use Case Diagram for Visitors capture

USE CASE DIAGRAM :

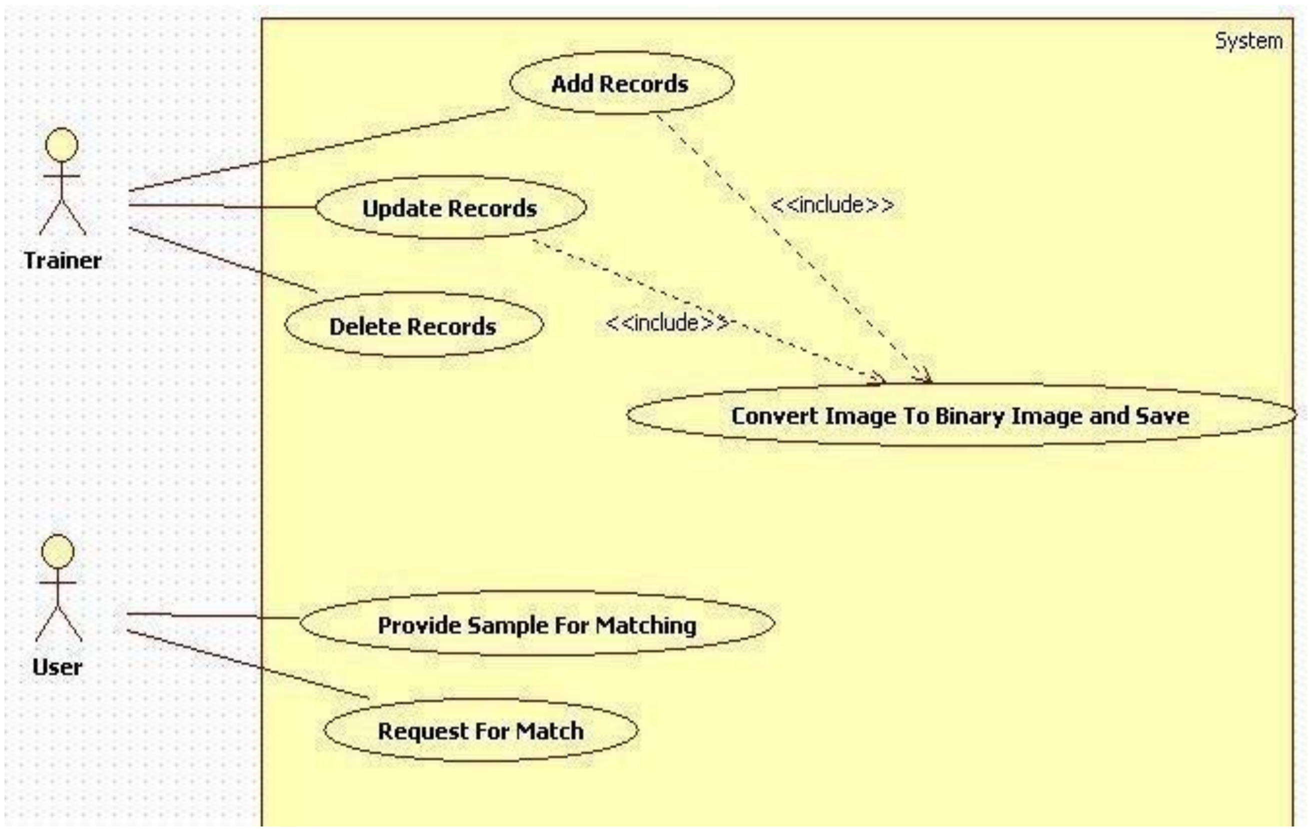


Fig 3: Use Case Diagram for Face Identification

USE CASE DIAGRAM :

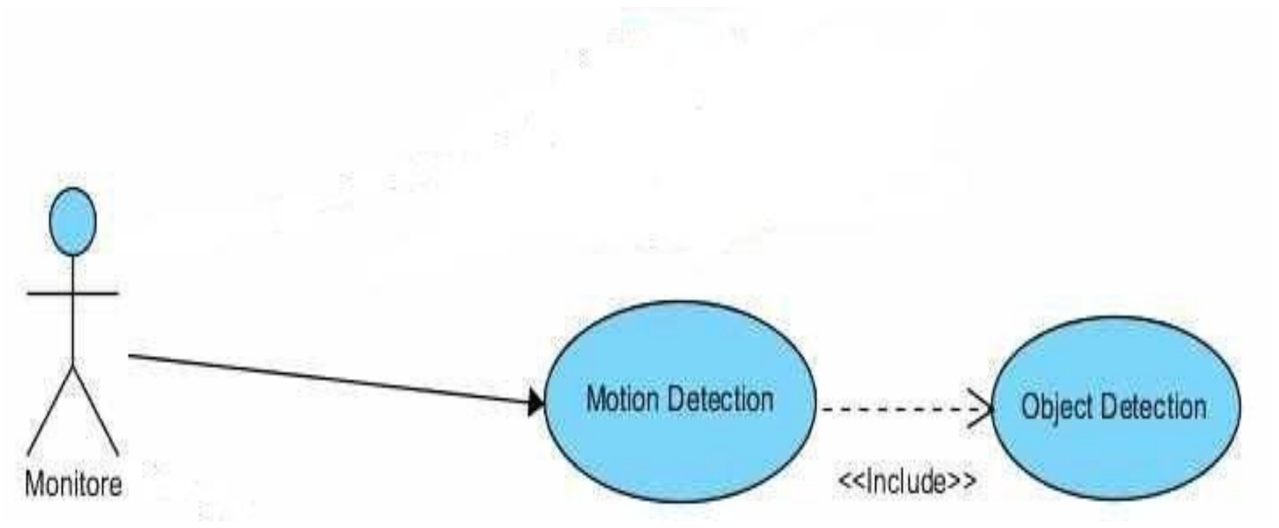


Fig 4: Use Case Diagram for Motion Detection

SEQUENCE DIAGRAM:

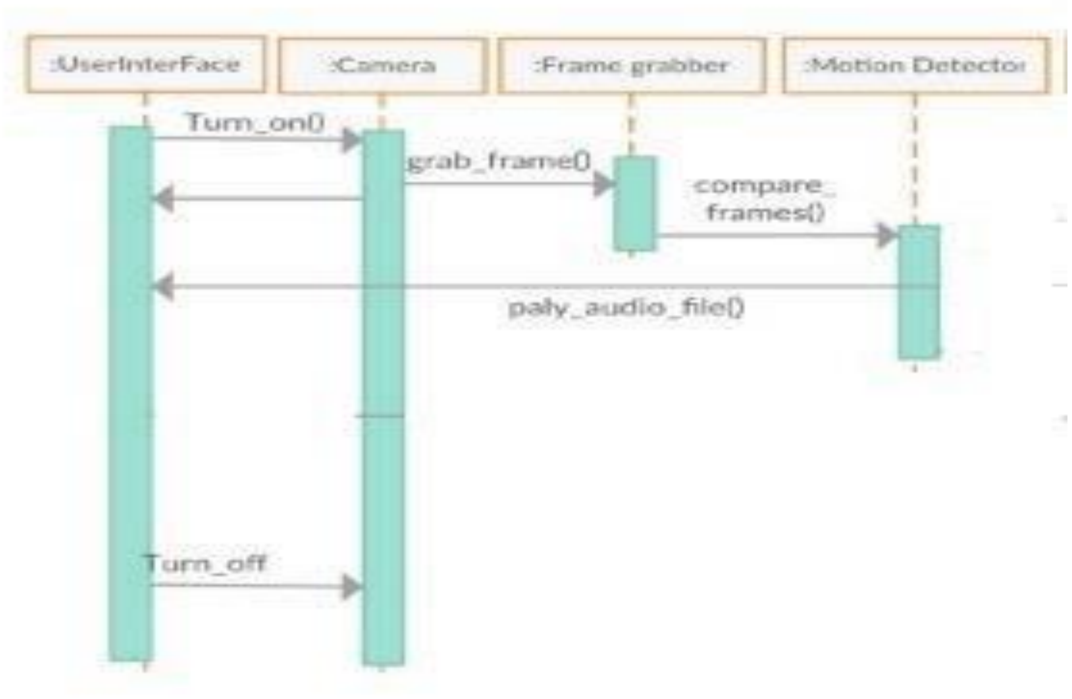


Fig 5 : Sequence Diagram for Motion Detection

SEQUENCE DIAGRAM:

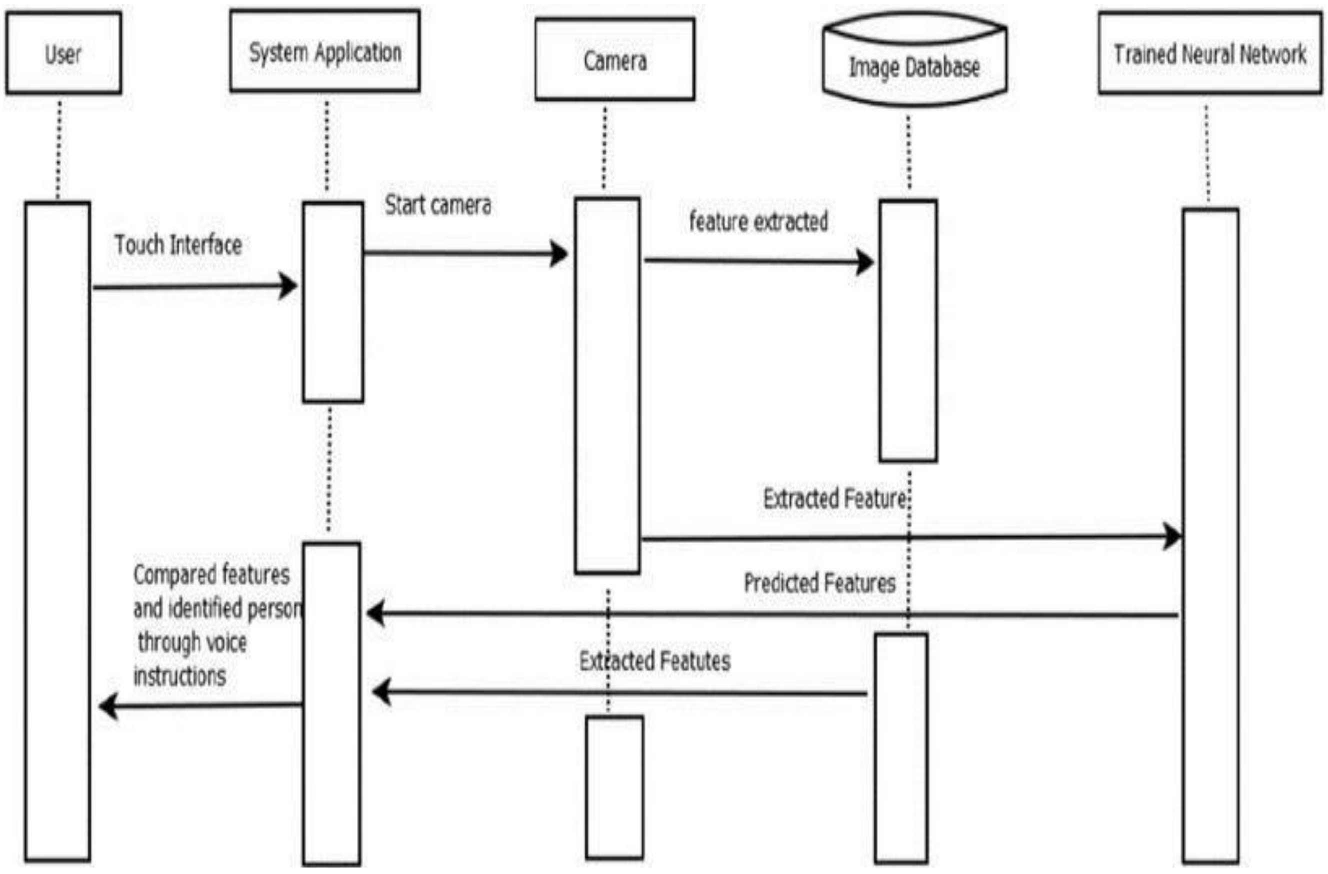


Fig 6: Sequence Diagram for Face Detection

SEQUENCE DIAGRAM:

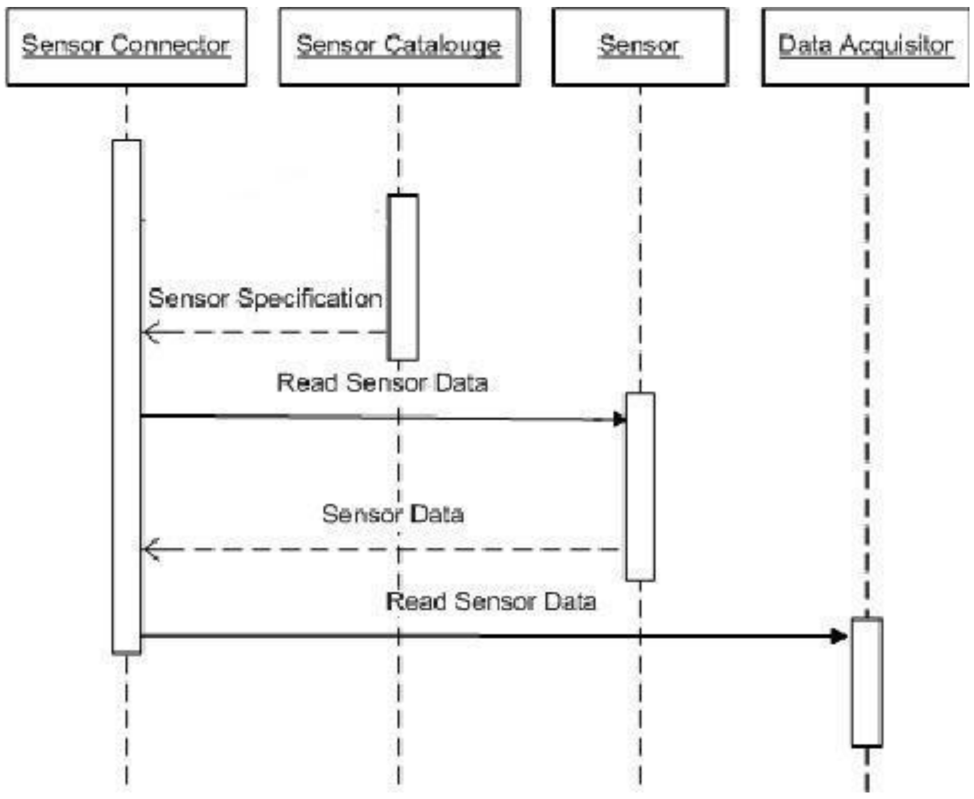


Fig 7: Sequence Diagram for Monitoring

ACTIVITY DIAGRAM:

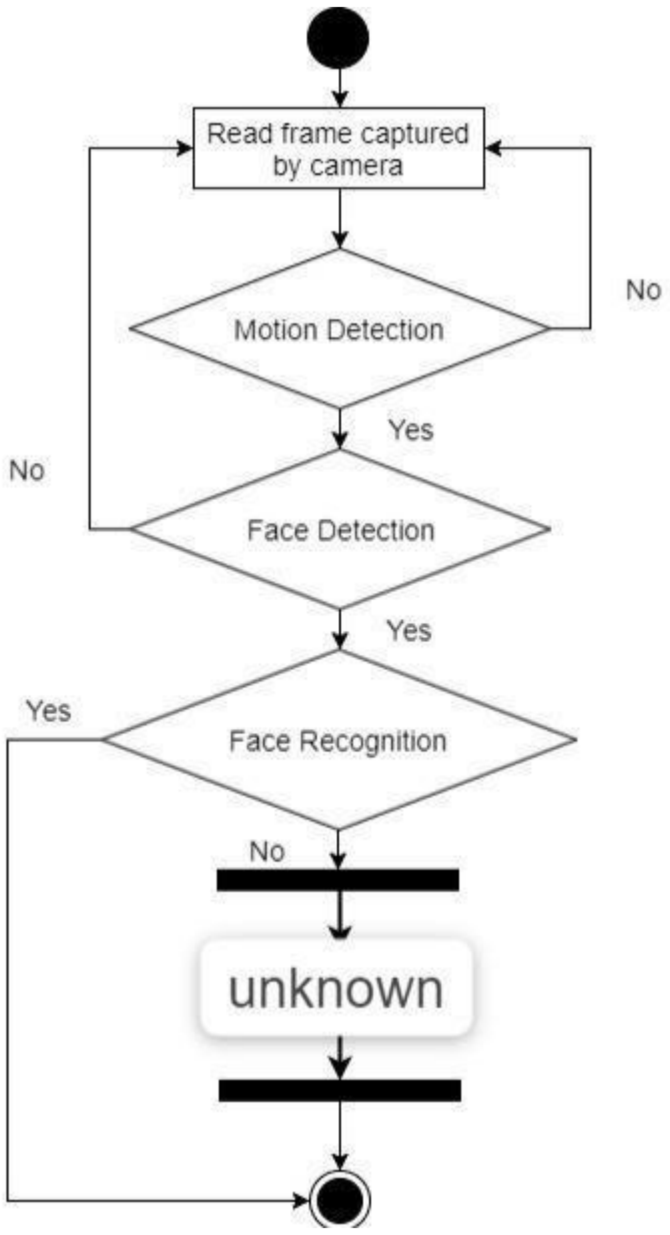


Fig 8: Activity diagram for face detection

ACTIVITY DIAGRAM:

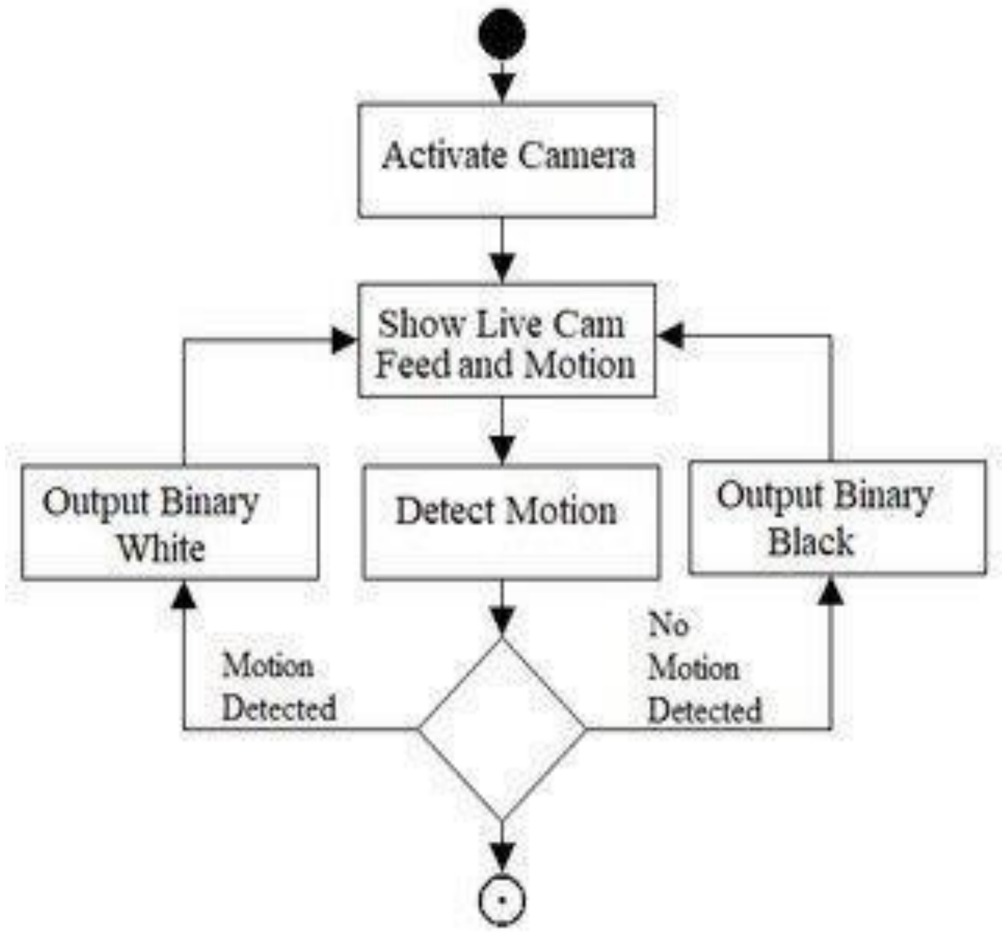


Fig 9 : Activity Diagram for Monitoring

ACTIVITY DIAGRAM:

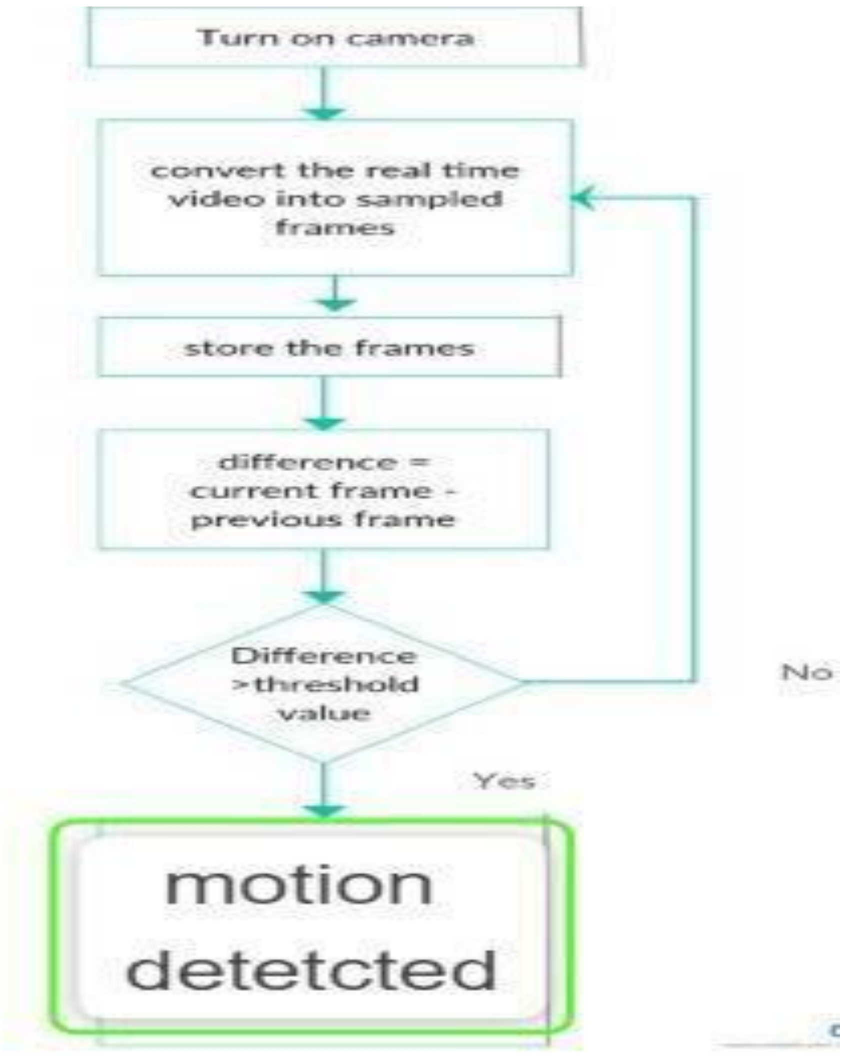

Turn on camera
convert the real time video into sampled frames
store the frames
difference = current frame - previous frame
Difference >threshold value
No
Yes
motion detetcted

Fig 10 : Activity Diagram for motion detection

CHAPTER 4

IMPLEMENTATION:

The application is implemented based on the following approaches:

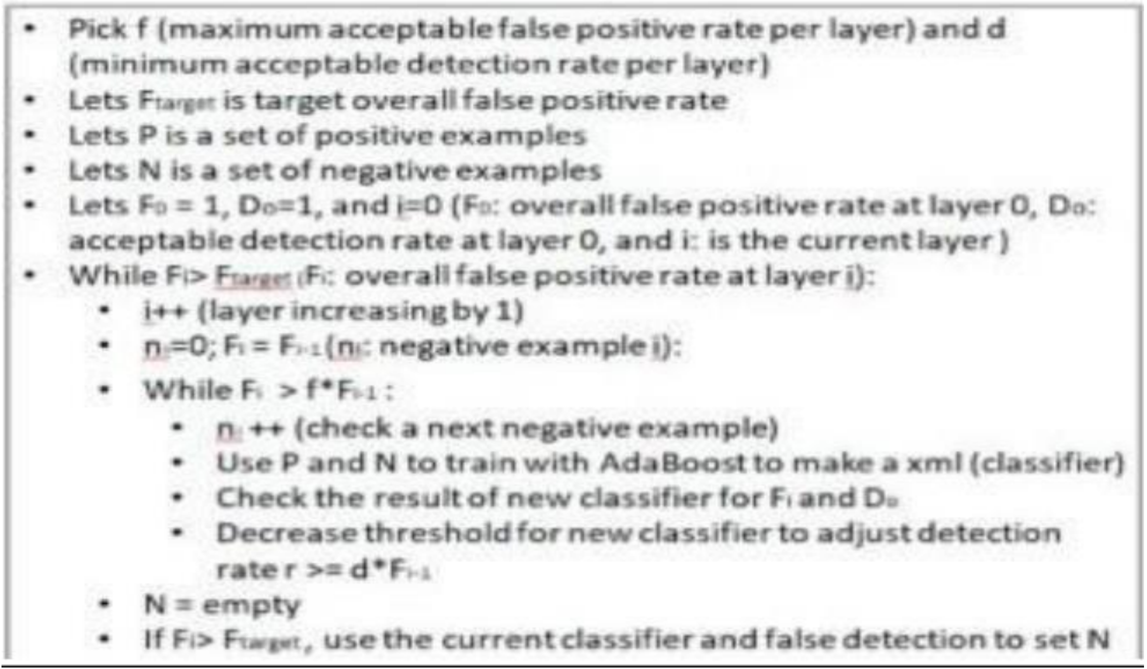
• Open CV :We have used this for Image processing like identifying faces, and Objects. It also performs operations like reading image , displaying image etc.

• Tkinter :We have used this frameworkto create GUI elements like buttons, elements , menu and data fields.

• Pillow :We have used Python Imaging Library (PIL),for opening, manipulating, and saving images.

• Haar Cascade Classifiers:We have used this object detection algorithm to identify faces in a image or a real time video.

Pseudo Code for Haar-cascade Algorithm:



```
• Pick f (maximum acceptable false positive rate per layer) and d
  (minimum acceptable detection rate per layer)
• Lets $F_{target}$ is target overall false positive rate
• Lets P is a set of positive examples
• Lets N is a set of negative examples
• Lets $F_0 = 1$, $D_0=1$, and $i=0$ ($F_0$: overall false positive rate at layer 0, $D_0$:
  acceptable detection rate at layer 0, and i: is the current layer )
• While $F_i > F_{target}$ ($F_i$: overall false positive rate at layer i):
    • i++ (layer increasing by 1)
    • $n_i=0$; $F_i = F_{i-1}$ ($n_i$: negative example i):
    • While $F_i > f*F_{i-1}$ :
        • $n_i$ ++ (check a next negative example)
        • Use P and N to train with AdaBoost to make a xml (classifier)
        • Check the result of new classifier for $F_i$ and $D_i$
        • Decrease threshold for new classifier to adjust detection
          rate $r >= d*F_{i-1}$
    • N = empty
    • If $F_i > F_{target}$, use the current classifier and false detection to set N
```

4.1 <u>Core functionality of the project:</u>

Steps for Face Prediction using Python and OpenCV are:

1. Create HAAR Cascade object using ‘CascadeClassifier’ function and ‘haarcascade_frontalface_default.xml’.

2. Read image using function ‘imread’ (or ‘read’ for video/ camera input) function.

3. Convert in gray scale using ‘cvtColor’ function.

4. Detect face using ‘detectMultiScale’ function.

5. Collecting data from HAAR Cascade and train data using LBPHFaceRecognizer_create() for identifying.

6. Observing the first frame using structural similarity index metric and comparing it with successive frames for analysis of any stolen objects in the camera vision range.

Important Code Snippets:

1. **Code snippet for opening window:**

```
btn_font = font.Font(size=25)
btn1 = tk.Button(frame1, text='Monitor', height=90, width=180, fg='green',command
= find_motion, image=btn1_image, compound='left') btn1['font'] = btn_font

btn1.grid(row=3, pady=(20,10))

btn2 = tk.Button(frame1, text='Rectangle', height=90, width=180, fg='orange',
command=rect_noise, compound='left', image=btn2_image) btn2['font'] =
btn_font
btn2.grid(row=3, pady=(20,10), column=3, padx=(20,5))

btn_font = font.Font(size=25)
btn3 = tk.Button(frame1, text='Noise', height=90, width=180, fg='green', command=noise,
image=btn3_image, compound='left')
btn3['font'] = btn_font
btn3.grid(row=5, pady=(20,10))

btn4 = tk.Button(frame1, text='Record', height=90, width=180,
fg='orange', command=record, image=btn4_image, compound='left')
btn4['font'] = btn_font
btn4.grid(row=5, pady=(20,10), column=3)


btn6 = tk.Button(frame1, text='In Out', height=90, width=180, fg='green', command=in_out,
image=btn6_image, compound='left')
btn6['font'] = btn_font
btn6.grid(row=5, pady=(20,10), column=2)

btn5 = tk.Button(frame1, height=90, width=180, fg='red', command=window.quit,
image=btn5_image)
btn5['font'] = btn_font
btn5.grid(row=6, pady=(20,10), column=2)

btn7 = tk.Button(frame1, text="identify", fg="orange",command=maincall, compound='left',
image=btn7_image, height=90, width=180)
btn7['font'] = btn_font
btn7.grid(row=3, column=2, pady=(20,10))

frame1.pack()
window.mainloop()
```

2. Code Snippet For Motion Detection:

```
   import cv2

   def noise():
 cap = cv2.VideoCapture(0)

 while True:
     _, frame1 = cap.read()
     _, frame2 = cap.read()

     diff = cv2.absdiff(frame2, frame1)
     diff = cv2.cvtColor(diff, cv2.COLOR_BGR2GRAY)

     diff = cv2.blur(diff, (5,5))
     _, thresh = cv2.threshold(diff, 25, 255, cv2.THRESH_BINARY)

     contr, _ = cv2.findContours(thresh,
cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

     if len(contr) > 0:
         max_cnt = max(contr, key=cv2.contourArea)
         x,y,w,h = cv2.boundingRect(max_cnt)
         cv2.rectangle(frame1, (x, y), (x+w, y+h), (0,255,0), 2)
         cv2.putText(frame1, "MOTION", (10,80), cv2.FONT_HERSHEY_SIMPLEX,
2, (0,255,0), 2)

     else:
         cv2.putText(frame1, "NO-MOTION", (10,80),
cv2.FONT_HERSHEY_SIMPLEX, 2, (0,0,255), 2)

     cv2.imshow("esc. to exit", frame1)

     if cv2.waitKey(1) == 27:
         cap.release()
         cv2.destroyAllWindows()
         break
```

3. **Code Snippet for Identify :**

```
def train():
print("training part initiated !")

recog = cv2.face.LBPHFaceRecognizer_create()

dataset = 'persons'

paths = [os.path.join(dataset, im) for im in os.listdir(dataset)]

faces = []
ids = []
labels = []
for path in paths:
labels.append(path.split('/')[-1].split('-')[0])

ids.append(int(path.split('/')[-1].split('-')[2].split('.')[0]))

faces.append(cv2.imread(path, 0))

recog.train(faces, np.array(ids))

recog.save('model.yml')

return


def identify():
cap = cv2.VideoCapture(0)

filename = "haarcascade_frontalface_default.xml"

paths = [os.path.join("persons", im) for im in os.listdir("persons")]
labelslist = []
for path in paths:
if path.split('/')[-1].split('-')[0] not in labelslist:
labelslist.append(path.split('/')[-1].split('-')[0])

print(labelslist)
recog = cv2.face.LBPHFaceRecognizer_create()

recog.read('model.yml')

cascade = cv2.CascadeClassifier(filename)

while True:
_, frm = cap.read()

gray = cv2.cvtColor(frm, cv2.COLOR_BGR2GRAY)
```

```
faces = cascade.detectMultiScale(gray, 1.4, 1)

for x,y,w,h in faces:
cv2.rectangle(frm, (x,y), (x+w, y+h), (0,255,0), 2)
roi = gray[y:y+h, x:x+w]

label = recog.predict(roi)

cv2.putText(frm, f"{labelslist[label[0]]}", (x,y), cv2.FONT_HERSHEY_SIMPLEX,
1, (0,0,255), 3)


cv2.imshow("identify", frm)

if cv2.waitKey(1) == 27:
cv2.destroyAllWindows()
cap.release()
break
```

4. **Code Snippet For Visitors capture:**

```
while True:
        _, frame1 = cap.read()
        frame1 = cv2.flip(frame1, 1)
        _, frame2 = cap.read()
        frame2 = cv2.flip(frame2, 1)

        diff = cv2.absdiff(frame2, frame1)

        diff = cv2.blur(diff, (5,5))

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

        _, threshd = cv2.threshold(gray, 40, 255, cv2.THRESH_BINARY)

        contr, _ = cv2.findContours(threshd,
cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

        x = 300
        if len(contr) > 0:
            max_cnt = max(contr, key=cv2.contourArea)
            x,y,w,h = cv2.boundingRect(max_cnt)
            cv2.rectangle(frame1, (x, y), (x+w, y+h), (0,255,0), 2)
            cv2.putText(frame1, "MOTION", (10,80), cv2.FONT_HERSHEY_SIMPLEX,
2, (0,255,0), 2)
```

5. Code Snippet For Monitoring:

```
def spot_diff(frame1, frame2):

frame1 = frame1[1]
frame2 = frame2[1]

g1 = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
g2 = cv2.cvtColor(frame2, cv2.COLOR_BGR2GRAY)

g1 = cv2.blur(g1, (2,2))
g2 = cv2.blur(g2, (2,2))

(score, diff) = structural_similarity(g2, g1, full=True)

print("Image similarity", score)

diff = (diff * 255).astype("uint8")
thresh = cv2.threshold(diff, 100, 255, cv2.THRESH_BINARY_INV)[1]

contors = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)[0]
contors = [c for c in contors if cv2.contourArea(c) > 50]

if len(contors):
for c in contors:

x,y,w,h = cv2.boundingRect(c)

cv2.rectangle(frame1, (x,y), (x+w, y+h), (0,255,0), 2)

else:
print("nothing stolen")
return 0

cv2.imshow("diff", thresh)
cv2.imshow("win1", frame1)
cv2.imwrite("stolen/"+datetime.now().strftime('%-y-%-m-%-d-%H:%M:%S')+".jpg",
frame1)
cv2.waitKey(0)
cv2.destroyAllWindows()

return 1
```

CHAPTER 5

TESTING :

1 . MOTION DETECTION:

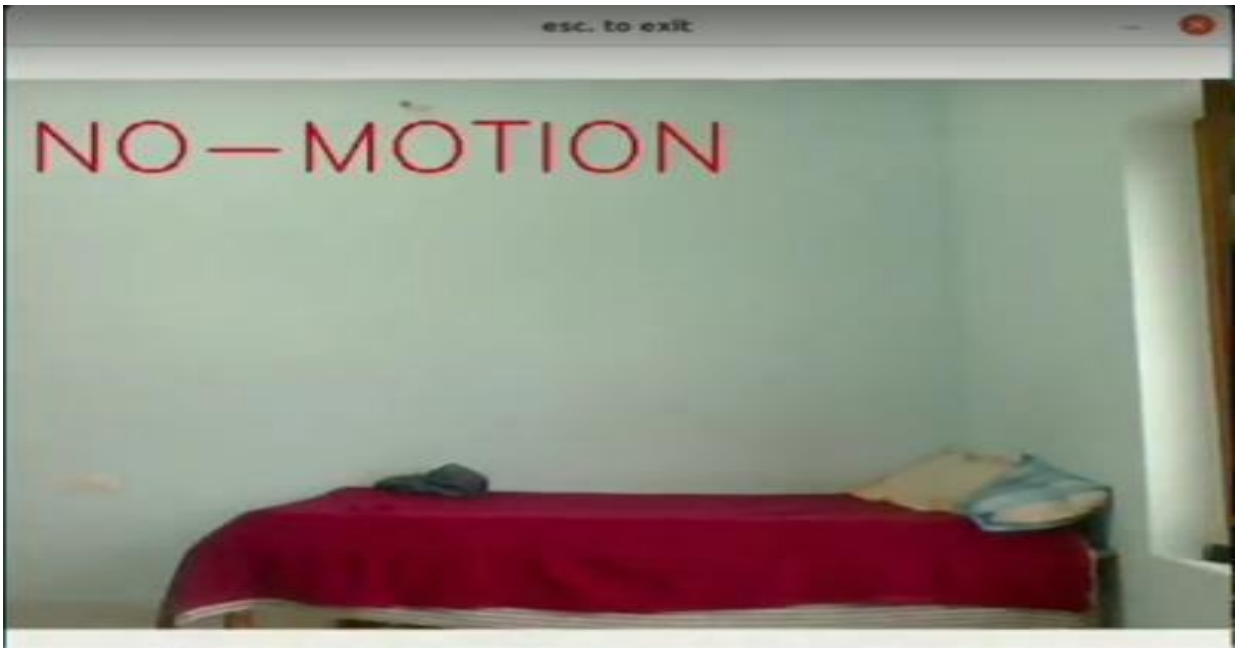


Fig 11 : No Motion Detection

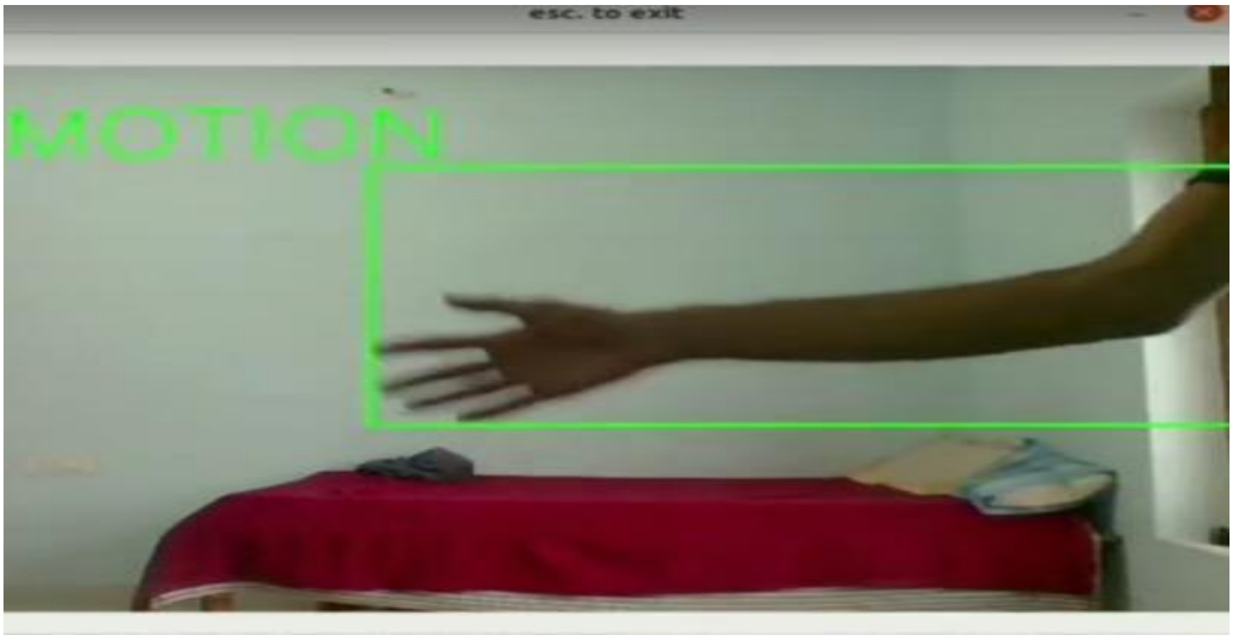


Fig 12 : Motion Detection

2. MONITORING:

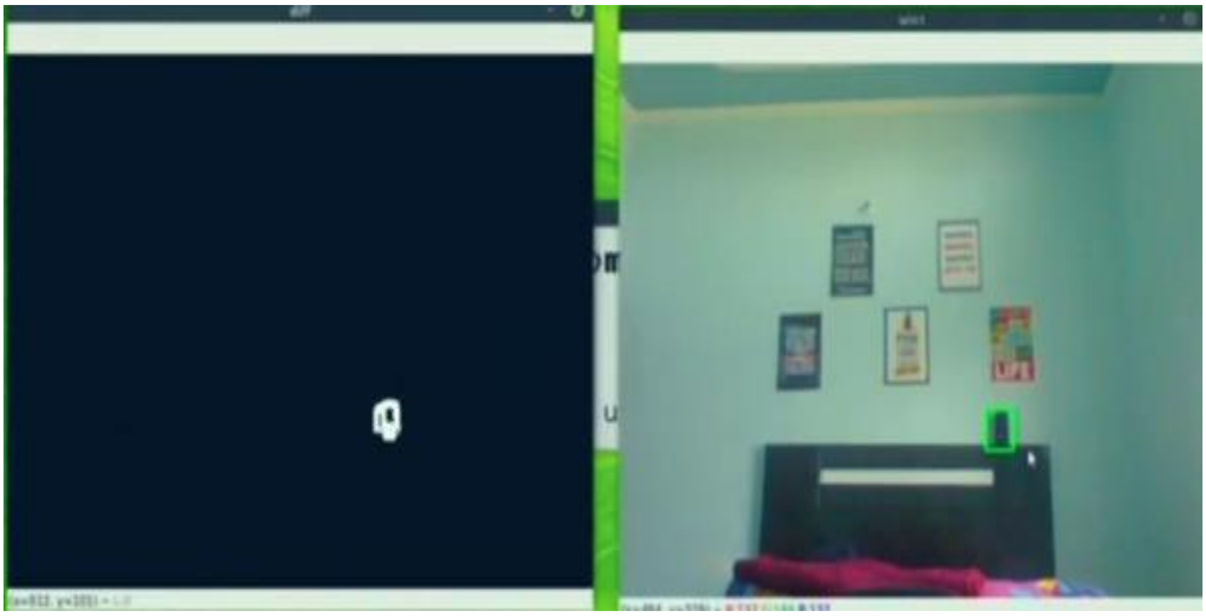


Fig 13 : Monitoring

3. FACE IDENTIFICATION:

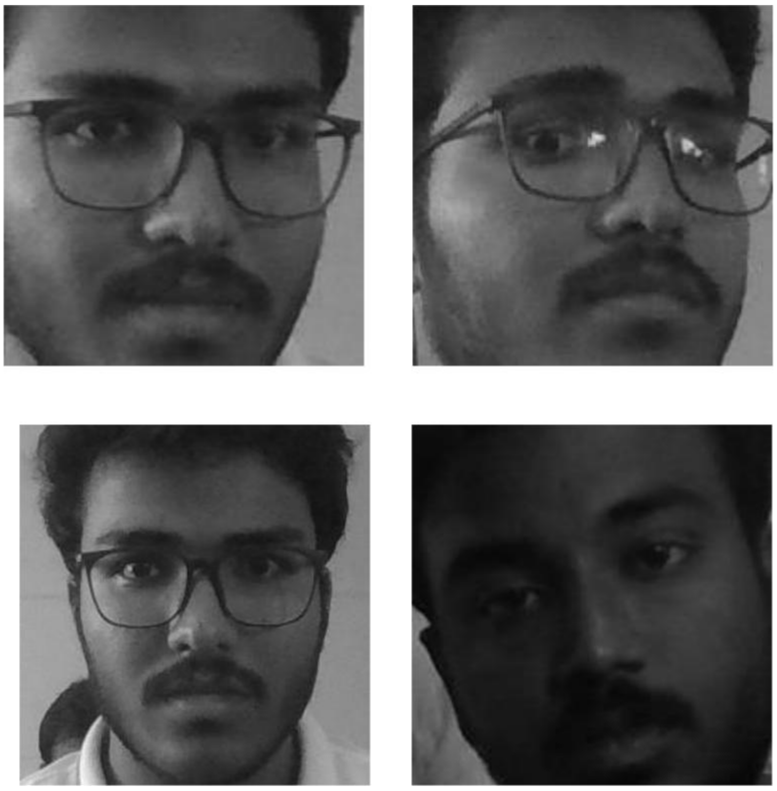


Fig 14 : Train Data

4. VISITORS CAPTURE:

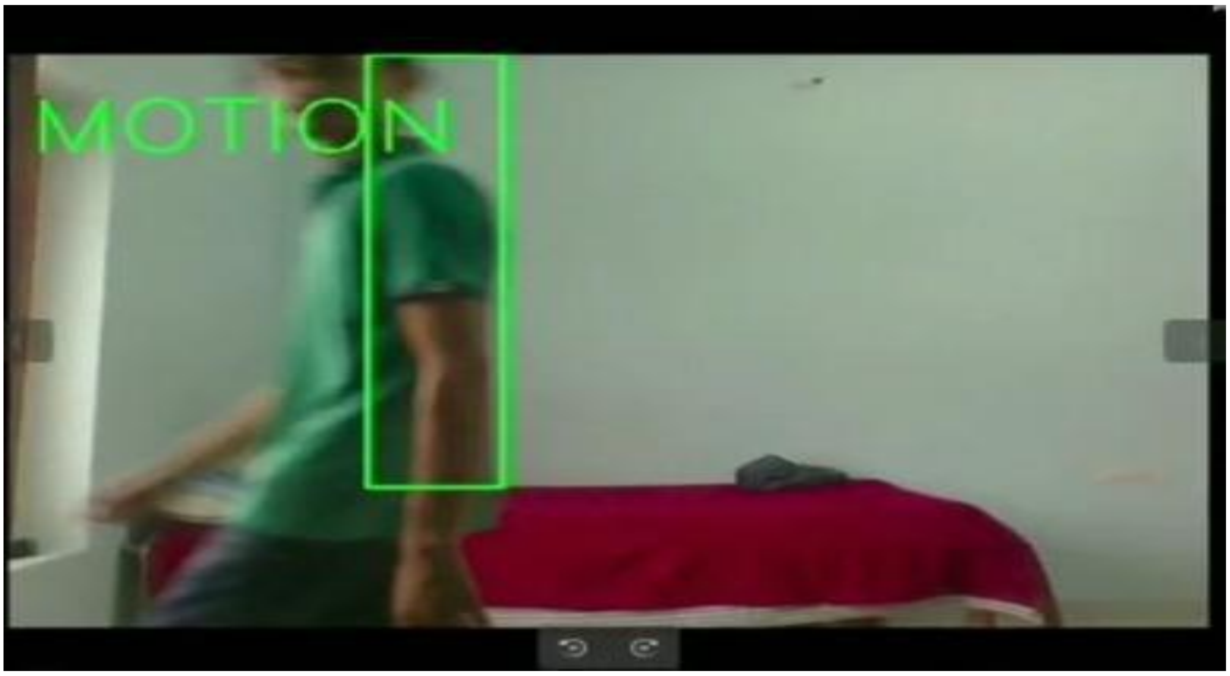


Fig 15 : Capturing the image of a visitor

CONCLUSION

This project is undertaken using machine learning and IQA algorithm and evaluates the results by using LBPH and SSIM. In our model we used haarcassade for the face detections, openCV for accessing the camera, beepy for the alerts, tkinter for GUI's, pillow for image processing and skimage for focus.

With the help of this project we can create a smart camera which can work without any human intervention and also performs various actions compared to a normal camera. The functonalities we achevied with the help of this are monitoring for anti-theft, identify for face identification, capture for alerts and capture, record for recording and motion for motion detection.

FUTURE ENHANCEMENTS

We can extend our research based on the technology improvements such as having the capability of small size but high processing power where this project can be broadly used.

We can extend this project by creating a Portable cctv, we can also add a in-built night vision capability and adding deep learning if we have a high power device.

We can also include more features such as Deadly weapon detection, accident detection, fire detection can also be included to this project.

Making a stand alone application with no requirements such as python language etc i.e also making a standalone device.

REFERENCES

[1] Sheikh, H. R., & Bovik, A. Anti-theft monitoring using IQA Algorithms. arXiv:2006.13846v2 [eess.IV] 29 Jun 2020.

[2] Rafael Padilla,Marly Costa. Evaluation of Haar Cascade Classifiers for Face Detection April 2012-Conference: ICDIP: International Conference on Digital Image Processing .At: Venice, Italy.Volume: 6.

[3] Naveenkumar ,Vadivel. OpenCV for Computer Vision Applications. March 2015-Conference: Proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15), March 20, 2015.At: Trichy.

[4] B.Kameswara Rao, Anusha Baratam. Face Identification Using machine learning. International Journal of Creative Research Thoughts (IJCRT) © 2020 IJCRT | Volume 8, Issue 5 May 2020 | ISSN: 2320-2882

[5] Pannaga Siri, Dr.Ganesh. Motion Detection Algorithms. International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 09 Issue: 02 | Feb 2022.

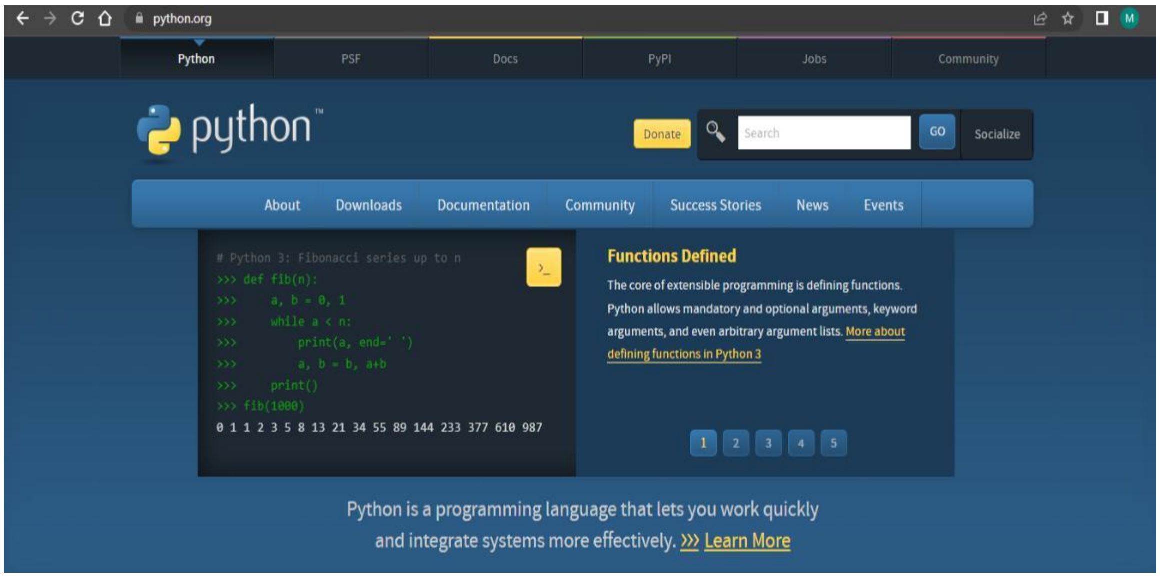
APPENDIX A – ABBREVIATIONS

1. RAM – Random Access Memory
2. IQA – Image Activity Assessment
3. LBPH – Local Binary Pattern Histogram
4. SSIM – Standard Schedules Information Manual
5. IDE – Integrated Development Environment
6. OpenCV – Open Sounrce Computer Vision Library
7. OS – Operating System
8. SSD – Solid State Drives
9. GB – GigaByte
10. CCTV – Closed Circuit Television

APPENDIX B – SOFTWARE INSTALLATION PROCEDURE

1. Python Installation:

Go to the official site to download and install python using Google Chrome or any other web browser or Click on the following link: **https://www.python.org/.**



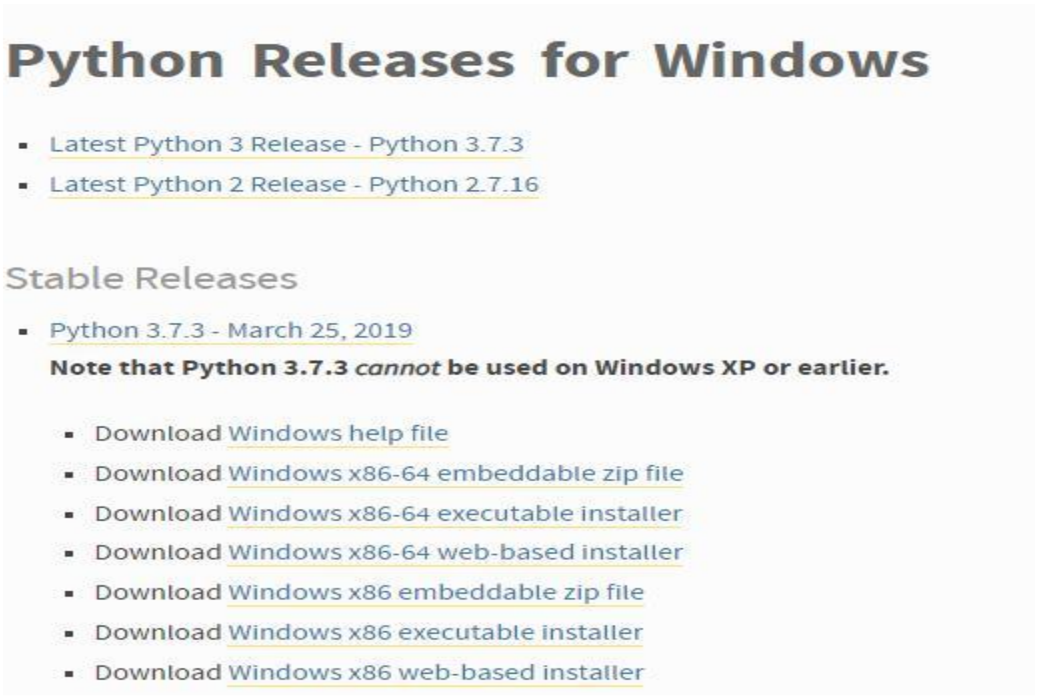
```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. More about defining functions in Python 3

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Fig 16 **:** Python Home Page



Python Releases for Windows

- Latest Python 3 Release - Python 3.7.3
- Latest Python 2 Release - Python 2.7.16

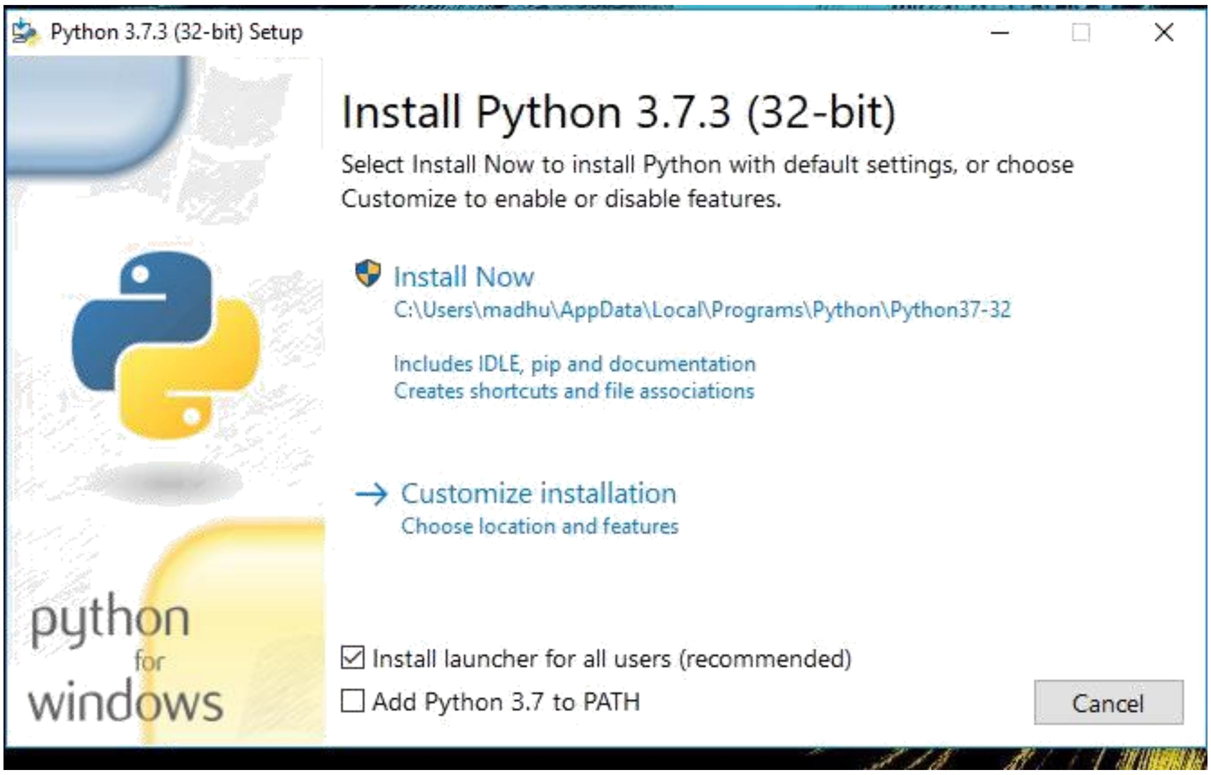
Stable Releases

- Python 3.7.3 - March 25, 2019

  Note that Python 3.7.3 *cannot* be used on Windows XP or earlier.

  - Download Windows help file
  - Download Windows x86-64 embeddable zip file
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 web-based installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86 executable installer
  - Download Windows x86 web-based installer

Fig 17 : Python Versions


Python 3.7.3 (32-bit) Setup

Install Python 3.7.3 (32-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

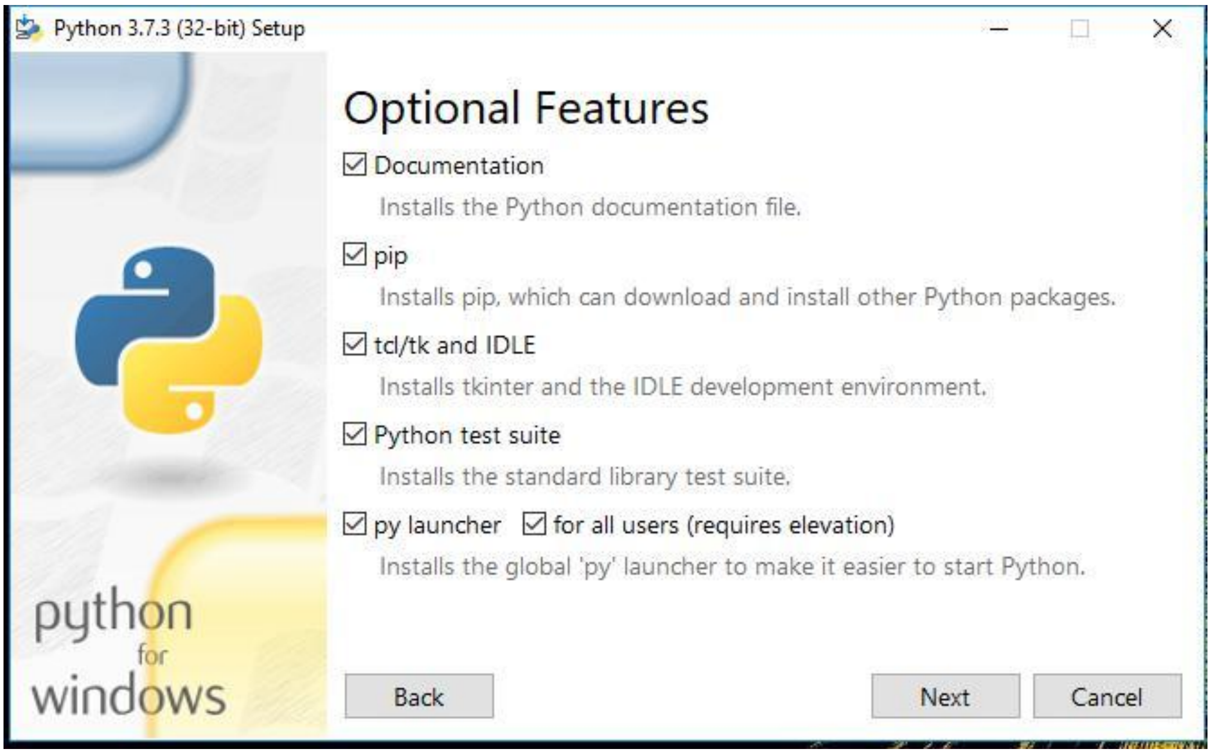
Install Now
C:\Users\madhu\AppData\Local\Programs\Python\Python37-32

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☐ Add Python 3.7 to PATH

Fig 18 : Run the Executable Installer


Python 3.7.3 (32-bit) Setup

Optional Features

☑ Documentation
Installs the Python documentation file.

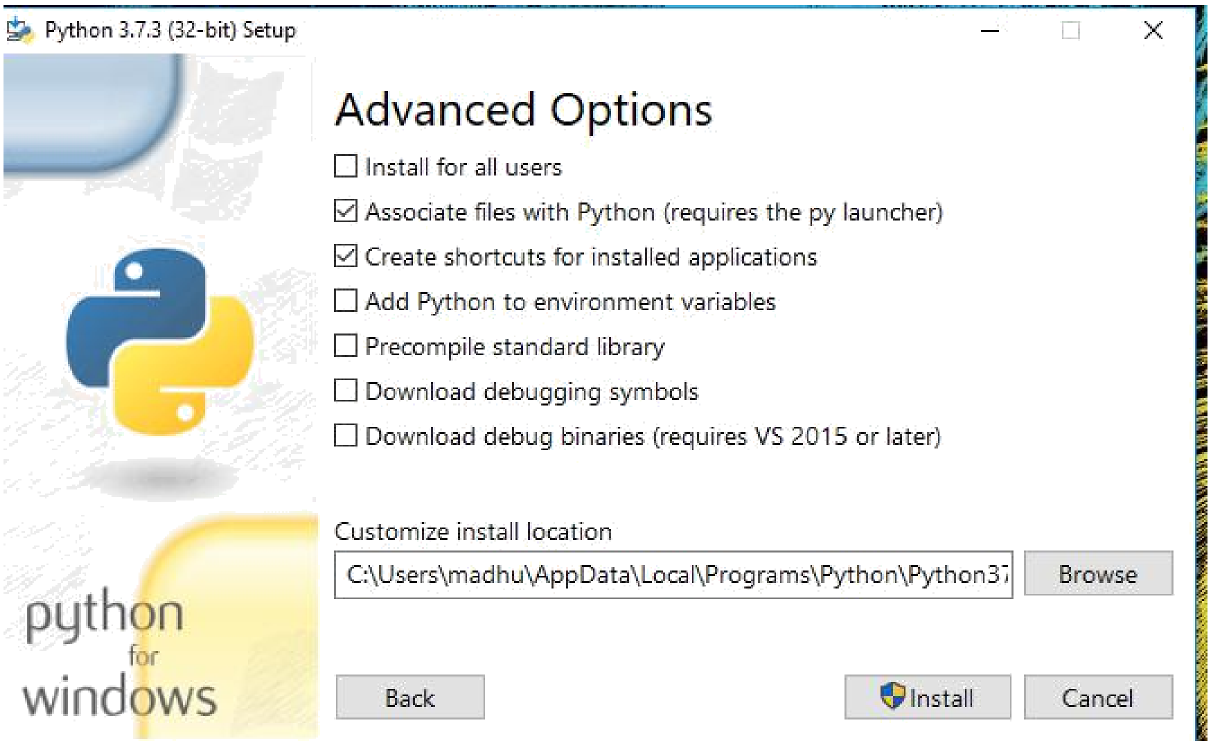
☑ pip
Installs pip, which can download and install other Python packages.

☑ tcl/tk and IDLE
Installs tkinter and the IDLE development environment.

☑ Python test suite
Installs the standard library test suite.

☑ py launcher ☑ for all users (requires elevation)
Installs the global 'py' launcher to make it easier to start Python.

Fig 19 : Features for python



Python 3.7.3 (32-bit) Setup

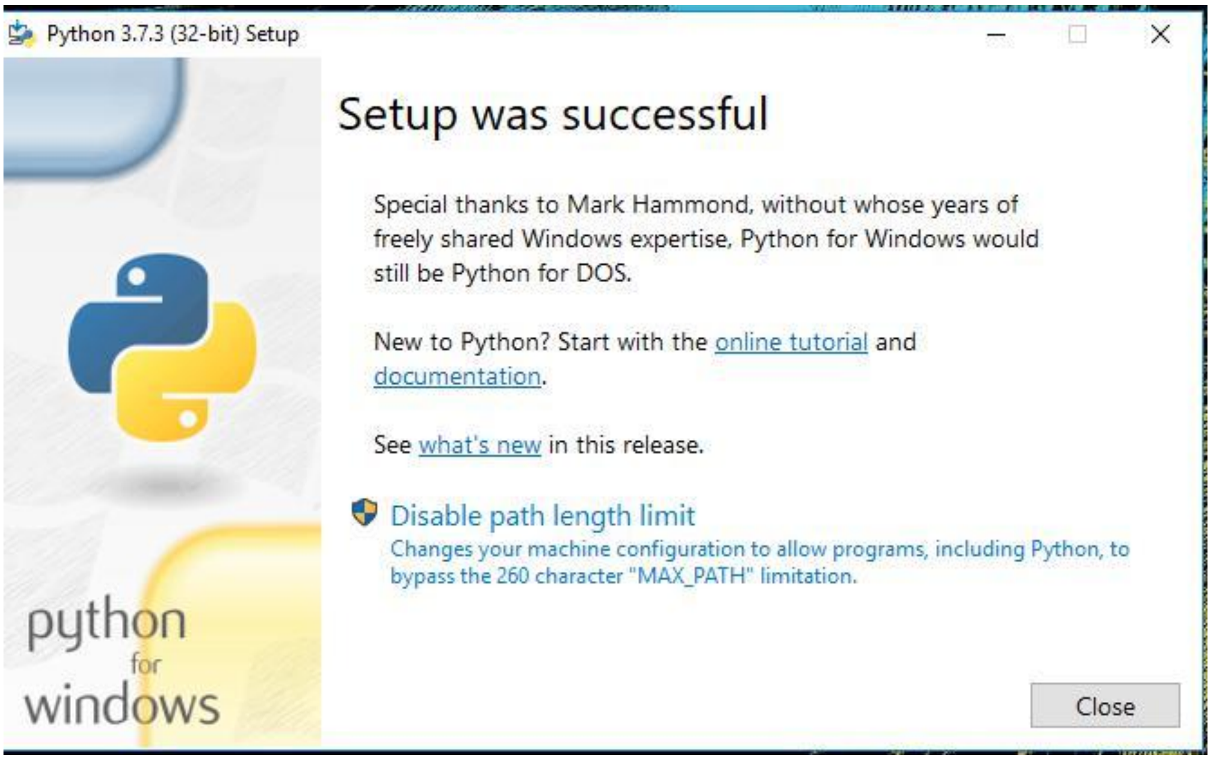
Advanced Options

- ☐ Install for all users
- ☑ Associate files with Python (requires the py launcher)
- ☑ Create shortcuts for installed applications
- ☐ Add Python to environment variables
- ☐ Precompile standard library
- ☐ Download debugging symbols
- ☐ Download debug binaries (requires VS 2015 or later)

Customize install location

C:\Users\madhu\AppData\Local\Programs\Python\Python37

Fig 20 : Advanced Options



Python 3.7.3 (32-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

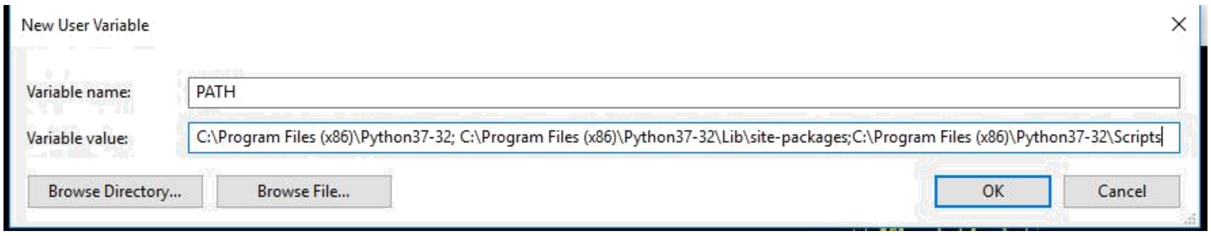
New to Python? Start with the online tutorial and documentation.

See what's new in this release.

Disable path length limit

Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

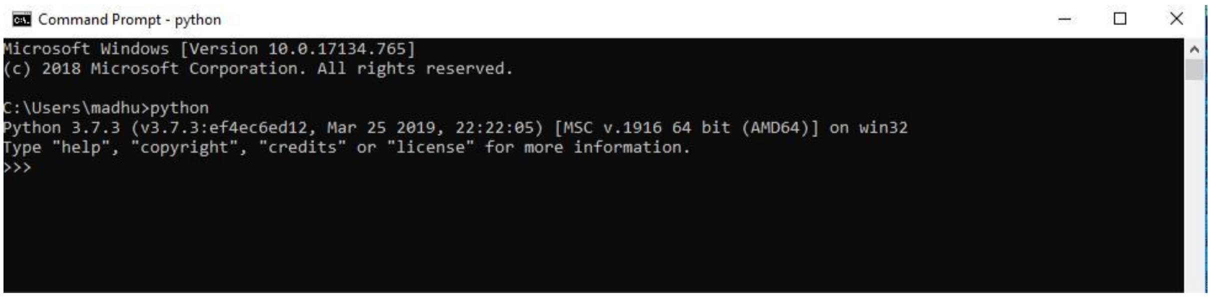
Fig 21 : Setup Successful


New User Variable

Variable name: PATH

Variable value: C:\Program Files (x86)\Python37-32; C:\Program Files (x86)\Python37-32\Lib\site-packages;C:\Program Files (x86)\Python37-32\Scripts

Fig 22 : Add python to environmental variables



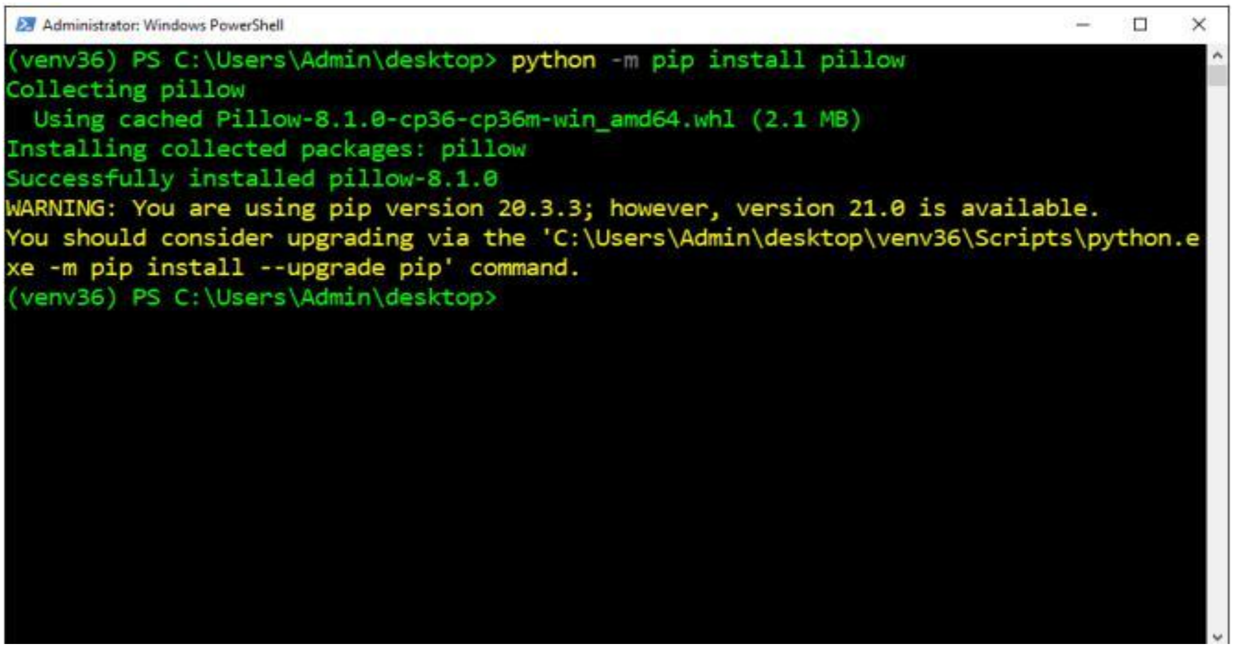
```
Microsoft Windows [Version 10.0.17134.765]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\madhu>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Fig 23 : Verification of python installation

2. Pillow Installation:

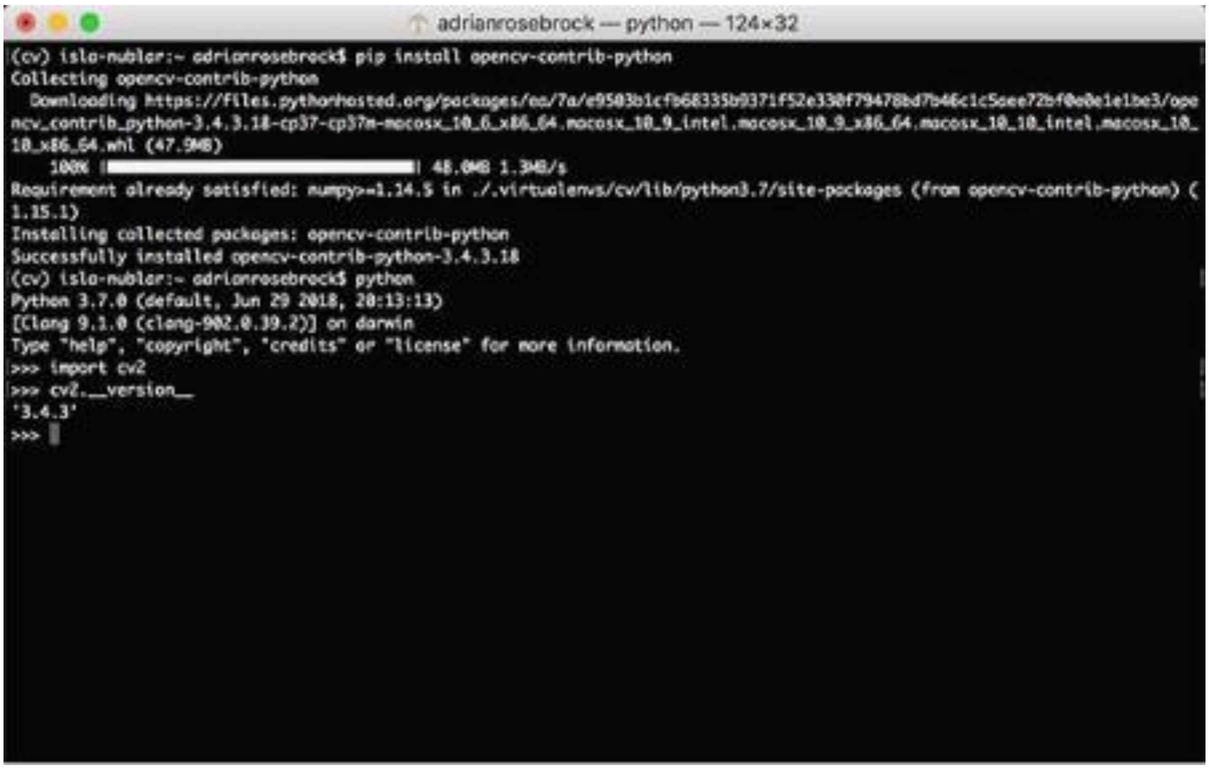
Open command prompt from start menu and Type the following command in command prompt.



```
(venv36) PS C:\Users\Admin\desktop> python -m pip install pillow
Collecting pillow
  Using cached Pillow-8.1.0-cp36-cp36m-win_amd64.whl (2.1 MB)
Installing collected packages: pillow
Successfully installed pillow-8.1.0
WARNING: You are using pip version 20.3.3; however, version 21.0 is available.
You should consider upgrading via the 'C:\Users\Admin\desktop\venv36\Scripts\python.e
xe -m pip install --upgrade pip' command.
(venv36) PS C:\Users\Admin\desktop>
```

Fig 24 : PIL installed

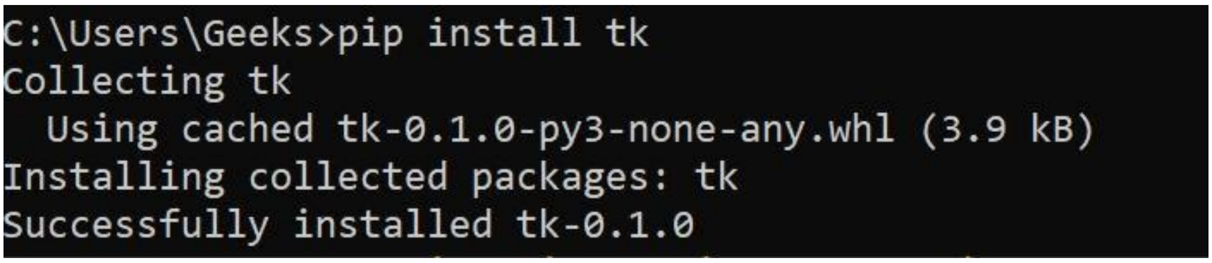
3. OpenCV Installation:



```
(cv) islo-nublar:~ adrianrosebrock$ pip install opencv-contrib-python
Collecting opencv-contrib-python
  Downloading https://files.pythonhosted.org/packages/ea/7a/e9503b1cfb68335b9371f52e330f79478bd7b46c1c5aee72bf0e0e1e1be3/ope
ncv_contrib_python-3.4.3.18-cp37-cp37m-macosx_10_6_x86_64.macosx_10_9_intel.macosx_10_9_x86_64.macosx_10_10_intel.macosx_10_
10_x86_64.whl (47.9MB)
    100% |████████████████████████████████| 48.0MB 1.3MB/s
Requirement already satisfied: numpy>=1.14.5 in ./.virtualenvs/cv/lib/python3.7/site-packages (from opencv-contrib-python) (
1.15.1)
Installing collected packages: opencv-contrib-python
Successfully installed opencv-contrib-python-3.4.3.18
(cv) islo-nublar:~ adrianrosebrock$ python
Python 3.7.0 (default, Jun 29 2018, 20:13:13)
[Clang 9.1.0 (clang-902.0.39.2)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'3.4.3'
>>>
```

Fig 25 : OpenCV Installed

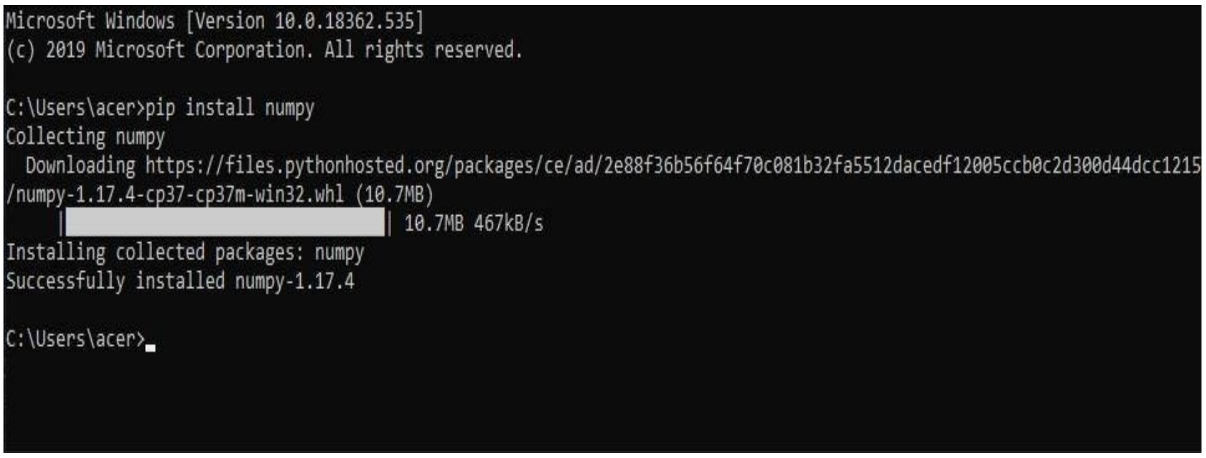
4. **Tkinter Installation:**



```
C:\Users\Geeks>pip install tk
Collecting tk
  Using cached tk-0.1.0-py3-none-any.whl (3.9 kB)
Installing collected packages: tk
Successfully installed tk-0.1.0
```

Fig 26 : Tkinter Installed

5. Numpy Installation:



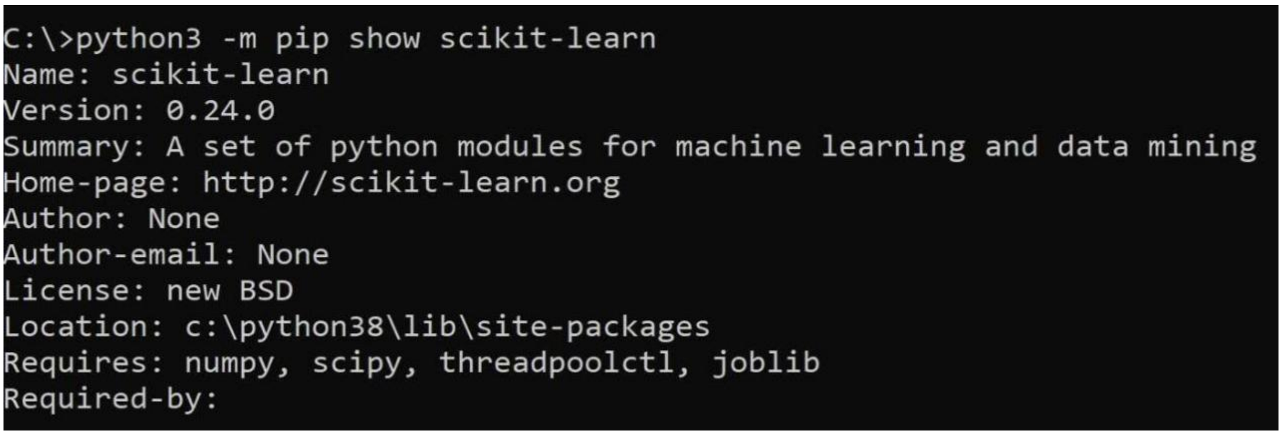
```
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\acer>pip install numpy
Collecting numpy
  Downloading https://files.pythonhosted.org/packages/ce/ad/2e88f36b56f64f70c081b32fa5512dacedf12005ccb0c2d300d44dcc1215
/numpy-1.17.4-cp37-cp37m-win32.whl (10.7MB)
     |████████████████████████████████| 10.7MB 467kB/s
Installing collected packages: numpy
Successfully installed numpy-1.17.4

C:\Users\acer>
```

Fig 27 : Numpy Installed

6 . Skimage Installation:

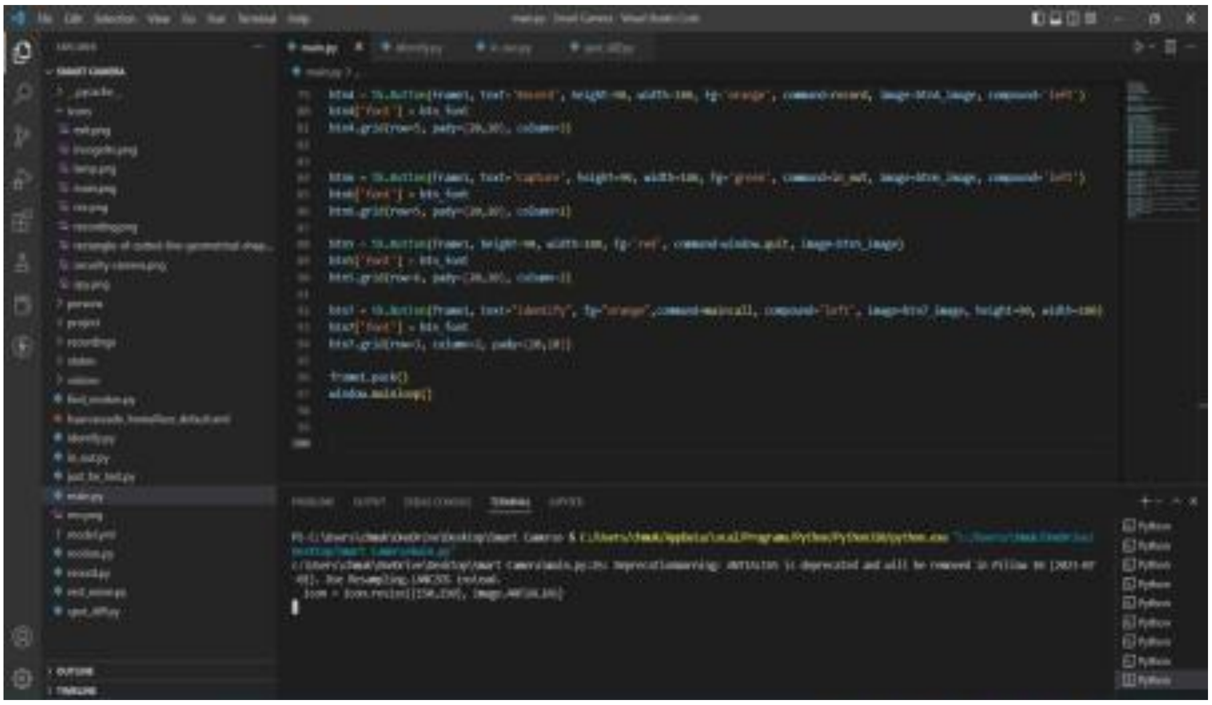


```
C:\>python3 -m pip show scikit-learn
Name: scikit-learn
Version: 0.24.0
Summary: A set of python modules for machine learning and data mining
Home-page: http://scikit-learn.org
Author: None
Author-email: None
License: new BSD
Location: c:\python38\lib\site-packages
Requires: numpy, scipy, threadpoolctl, joblib
Required-by:
```

Fig 28 : Skimage Installed

APPENDIX C – SOFTWARE USAGE PROCESS

1. **Starting the application:**



```
btn4 = tk.Button(frame1, text='Record', height=90, width=180, fg='orange', command=record, image=btn4_image, compound='left')
btn4['font'] = btn_font
btn4.grid(row=5, pady=(20,10), column=3)


btn6 = tk.Button(frame1, text='Capture', height=90, width=180, fg='green', command=in_out, image=btn6_image, compound='left')
btn6['font'] = btn_font
btn6.grid(row=5, pady=(20,10), column=2)

btn5 = tk.Button(frame1, height=90, width=180, fg='red', command=window.quit, image=btn5_image)
btn5['font'] = btn_font
btn5.grid(row=6, pady=(20,10), column=2)

btn7 = tk.Button(frame1, text="identify", fg="orange",command=maincall, compound="left", image=btn7_image, height=90, width=180)
btn7['font'] = btn_font
btn7.grid(row=3, column=2, pady=(20,10))

frame1.pack()
window.mainloop()
```

```
PS C:\Users\[illegible]\OneDrive\Desktop\Smart Camera> & C:/Users/[illegible]/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/[illegible]/OneDrive/Desktop/Smart Camera/main.py"
c:\Users\[illegible]\OneDrive\Desktop\Smart Camera\main.py:[illegible]: DeprecationWarning: ANTIALIAS is deprecated and will be removed in Pillow 10 (2023-07-01). Use Resampling.LANCZOS instead.
  icon = icon.resize((150,150), Image.ANTIALIAS)
```

Fig 29 : Application Starts

2. **Opening Window:**

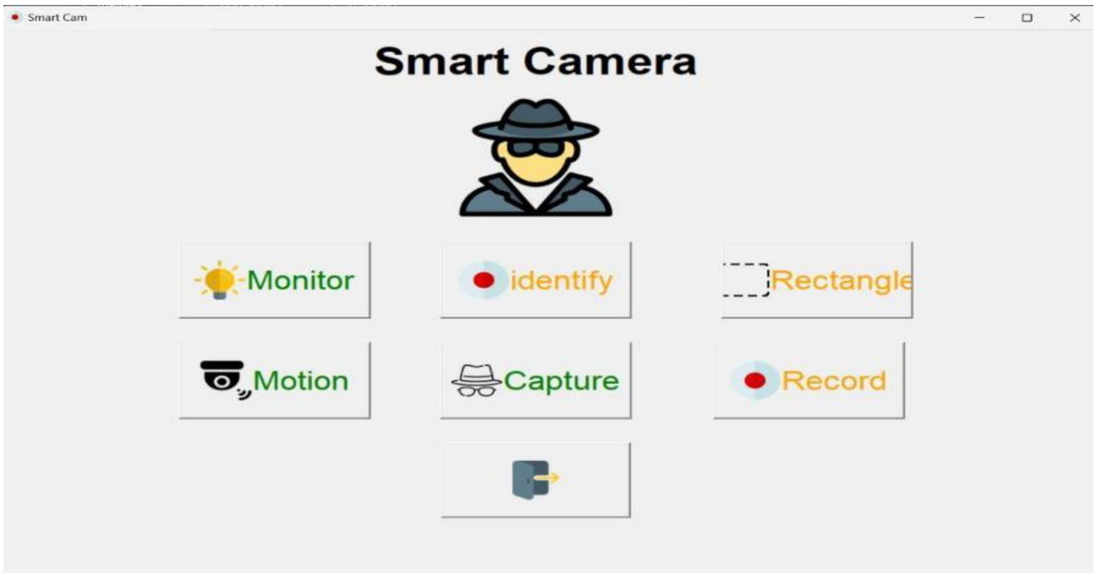


Fig 30 : Landing Window

3. Camera Starts:

Adjust your face accordingly so that the camera can detect.

Fig 31: Camera Starts