

SMART CCTV TKINTER

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Science degree in
Computer Science

by

S. Jayashree (3929038)

S. Mathana (39290063)



DEPARTMENT of COMPUTER SCIENCE SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI - 600 119**

MARCH - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of S.JAYASHREE (39290038)
And S.MATHANA (39290063) who carried out the project entitled“ SMART CCTV TKINTER
” under my supervision from November 2021 _to April 2022.

Internal Guide

(Dr.M.SELVI,M.E.,Ph.,D)

Head of the Department

Dr . L.LAKSHAMANAN , M.E., Ph..D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I ,S.JAYASHREE (Reg.no. 39290038) hereby declare that the Project Report entitled "SMART CCTV TKINTER" done by me under the guidance of Dr ..M.SELVI.

is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in COMPUTER SCIENCE.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T . Sasikala M.E., Ph.D, Dean, School of Computing** **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr./Mr./Ms for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

TABLE OF CONTENTS

S . NO	TITLE	PAGE No.
	ABSTRACT	
	LIST OF FIGURES	
1.	INTRODUCTION	1
2.	LITERATURE SURVEY	
2.1	SMART SURVEILLANCE SYSTEM	
2.2	BACKGROUND VIDEO SURVEILLANCE	2
2.3	BACKGROUND AND PROGRESS OF DETECTION TOOL	
2.4	LITERATURE REVIEW NOVEL DIRECTION	3
3.	AIM AND SCOPE OF PRESENT INVESTIGATION	
3.1	AIM AND SCOPE	4
3.2	SYSTEM REQUIREMENT	
3.3	SOFTWARE REQUIREMENT	
3.4	HARDWARE REQUIREMENT	5
3.5	PYTHON	
3.6	APPLICATION OF PYTHON PROGRAMMING	7
4.	EXPERIMENTAL OR MATERIAL METHOD	
4.1	DESIGN METHODOLOGY	9
	4.1.1. EXISTING SYSTEM	
	4.1.2. PROPOSED SYSTEM	

4.2	MODULE DESCRIPTION	10
	4.2.1. ANOMALY MODULE	
	4.2.2.OBJECT	
	MODULE 4.2.3.FACE MODULE	
4.3	THE ANALYSIS COMPOUND	12
	4.3.1.ROLE MODULE	
	4.3.2.DESCRPTION MODULE	
4.4	THE INTERACTION BETWEEN DETECTION AND ANALYSIS	
	COMPOUNDS	13
4.5	ARCHITECTURE DIAGRAM	15
5.	RESULT AND PERFORMANCE ANALYSIS	
5.1	APP FRONT LAYOUT	16
5.2	DETECTION(MOTION AND NON MOTION)	16-19
6.	CONCLUSION AND FUTURE WORK	20
7.	REFRENC	21-23
8.	APPENDIX	23-36
8.1	SOURCE CODE	

Chapter1

1.introduction

Chapter Summary:

Video surveillance is of vital importance to our public, company, and home safety. Most existing commercialised surveillance systems can only detect

Goals and Challenges

The primary goal of this thesis is to study and improve the performance of current methods applied in novel domains. Also, the thesis also explores event description/understanding area.

Namely, the primary goal is to extend the surveillance system to different domains as pioneer works. The ability to work under different domain is also a significant advantage for smartvideo surveillance. The investigation in this direction is necessary and can contribute to the related research communities. These goal will be investigating the event understanding and description functions for video surveillance system. This will allow better human- machine interaction and step further to the direction of artificial intelligence.

An advance of miniaturization and lower the cost of cameras (Raspberry pi 3) have preferred the implementation of large-scale networks of the camera. This increasing number of cameras could permit novel signal processing applications which employ multiple sensors in extensive areas.

This is a tkinter gui app for smart camera. You just need to run main.py file inorder to run full app. You would need : opencv tkinter.

Installed to run these scripts properly.

- . We have implemented a system which provides both face detection and face recognition with the help of Raspberry pi 3 which is a credit card sized minicomputer and a Pi camera which is made especially for the raspberry pi 3.
- Thus, when dealing with the real-time image processing, Open source computer vision (openCV) software,a powerful library of image processing tools, is a good choice.
- With the help of a smart surveillance system, we have achieved a system that can record the event, detect and recognize the person

Chapter2

2.LiteratureReview

2.1 Smart Surveillance System

This chapter provides related information on video surveillance, object detection, anomaly detection, and face

Surveillance cameras are a part of our daily life in modern cities. Traffic cameras prevent/capture dangerous behaviours and keep the road safer; Shop cameras can stop/catch stealing or robbery; In the public environments such as the airport or company building, cameras are running somewhere you may not even notice. Even in your house, a surveillance system can be installed to protect the property.

In its many forms, video surveillance has been serving us for various purposes for a long time. This industry is about to change due to computer vision related research such as object/face/event detection, video event analysis and description [113].

2.2 Background of Video Surveillance Industry

Surveillance is the mechanism of recording what is happening in the form of audio/video data. Video surveillance has been invented for more than 70 years.

The Australian Concise Oxford Dictionary defines *surveillance* as “Close observation, especially of a suspected spy or criminal”. The dictionary further states the origin as Early 19th century: from French, from *sur-over* + *veiller* watch (from Latin *vigilare* keep watch).

Naturally, this word is closely related to safety, security, privacy and

observation. Surveillance can be served for positive or negative purposes. It's one of the most important elements in Hollywood FBI or spies movies in the form of different fancy audio/video recording devices. In this thesis, the scenario is limited to video surveillance for public security purpose.

Traditionally, video surveillance is also known as closed-circuit television (CCTV).

Although CCTV is defined in the perspective of hardware. In many decades, the video surveillance

The Australian Concise Oxford Dictionary defines *surveillance* as “Close observation, especially of a suspected spy or criminal”. The dictionary further states the origin as Early 19th century: from French, from *sur-over* + *veiller* watch (from Latin *vigilare* keep watch).

Naturally, this word is closely related to safety, security, privacy and

observation. Surveillance can be served for positive or negative purposes. It's one of the most important elements in Hollywood FBI or spies movies in the form of different fancy audio/video recording devices. In this thesis, the scenario is limited to video surveillance for public security purpose.

Traditionally, video surveillance is also known as closed-circuit television (CCTV).

Although CCTV is defined in the perspective of hardware. In many decades, the video surveillance system is

2.3 Background and Progress of Related Detection Tools

The techniques in automatic video surveillance covers a broad range. Among the many topics, the essential functions are the detection abilities. In this section, we only introduce the background knowledge of detection topics that are related to this thesis. Literature overviews are also provided to better understand the progress of each topic.

After the discussion on the history and importance of video surveillance, in this section, we will go through the video surveillance detection functions that are related to this thesis.

The

background knowledge and progress of the related detectors will be introduced together with a literature overview on each topic.

Research topics including object detection/recognition, face detection/alignment, and anomaly detection will be introduced. Note that this section provides a literature overview to help illustrate the necessary knowledge on each topic.

2.4 Literature Review on Novel Directions

Previous sections have introduced the related techniques of modern video surveillance systems

withabroadrangeofliteratureoverview.Inthissection,adeeperandmorespecificliteraturereview willbe given on the two main directions of this thesis: 1) video surveillance in novel domains; 2) eventunderstandinganddescription.

Note that there are only a few papers working in these two directions so this section is relativelyshortcomparestoprevioussections.Also,fromthetablesbelow,asarevisitofourproposedresearchquestions,wefindtheanswersfromtheliteraturearenotsatisfying.

CHAPTER 3

3.1 SCOPE AND OBJECTIVES:

- enhances the safety of the area.
- records archive footage for later reference.
- Cameras and monitors enable you to view events live.
- Allows you to keep an eye on what's going on in and around your business.
- CCTV features included creating a safe environment, protection, legal compliance and audit, management and support, deterrence, investigation and evidence, all of which were considered at each stratum levels; however, with difference in their meaning.
- detect all the irregular movements within the area.
- Increase the intelligence of the video surveillance system.
- select the suspicious ones conduct in-depth observation.
- identification of the targets.

3.2 SYSTEM REQUIREMENTS:

SOFTWARE REQUIREMENT:

- Operating System.
- Windows . Coding Language.
- Python Library.
- OpenCV

HARDWARE REQUIREMENT:

- Hard Disk: 500GB and above.
- RAM: 4GB and above.
- Processor: 13 and above.
- Web Camera.

3.3 PHYTHON:

Python Language:

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc. Python programming is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability & this class will teach you python from basics.

Python Characteristics:

- It provides rich data types and easier to read syntax than any other programming languages
- It is a platform independent scripted language with full access to operating system

API's

- Compared to other programming languages, it allows more run-time flexibility • It includes the basic text manipulation facilities of Perl and Awk
- A module in Python may have one or more classes and free functions
- Libraries in Python are cross-platform compatible with Linux, Macintosh, and Windows
- For building large applications, Python can be compiled to byte-code
- Python supports functional and structured programming as well as OOP
- It supports interactive mode that allows interacting Testing and debugging of snippets of code
- In Python, since there is no compilation step, editing, debugging and testing is fast.

Applications of Python Programming:

Web Applications:

You can create scalable Web Apps using frameworks and CMS (Content ManagementSystem) that are built on Python. Some of the popular platforms for Scientific and Numeric Computing:

Creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS. Sites like Mozilla,Reddit, Instagram and PBS are written in Python.

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: Earthy for earth science, Astray for Astronomy and soon. Also, the language is heavily used in machine learning, data mining and deep learning.

Creating software Prototypes:

Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must. However, Python is a great language for creating prototypes. For example: You can use Pygmy (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

Good Language to Teach Programming:

Python is used by many companies to teach programming to kids and newbies. It is a good language with a lot of features and capabilities. Yet, it's one of the easiest languages to learn because of its simple easy-to-use syntax.

OpenCV Package

Python is a general-purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability.

CHAPTER 4 :EXPERIMENTAL OR MATERIAL METHOD

Anextensiveframeworkcoveringmultipletopicsforonesinglepurpose:abetterandsmartersu
Rveillanceprototype.

To understand the structure of the framework, the connection map between Chapters are Shownagainbelow. As shown in Figure 4.1, the two principal elements are the detection and analysis component.First, The system will detect event/object of interest inthe video. This will be done by the detection component.When the detection component has gatherednecessaryinformation,itwillpassthemes-sage to the analysis component for higher level analysis such as role analysis, event understanding, and video description. Ideally, the system provides meaningful output such as an alert when there are important /abnormal events .Also, the system will generate the textual description of the processedvideo. Also,The frameworkcanbemodifiedforvarious purposesbydisablingorenablingthe sub-modules which will be

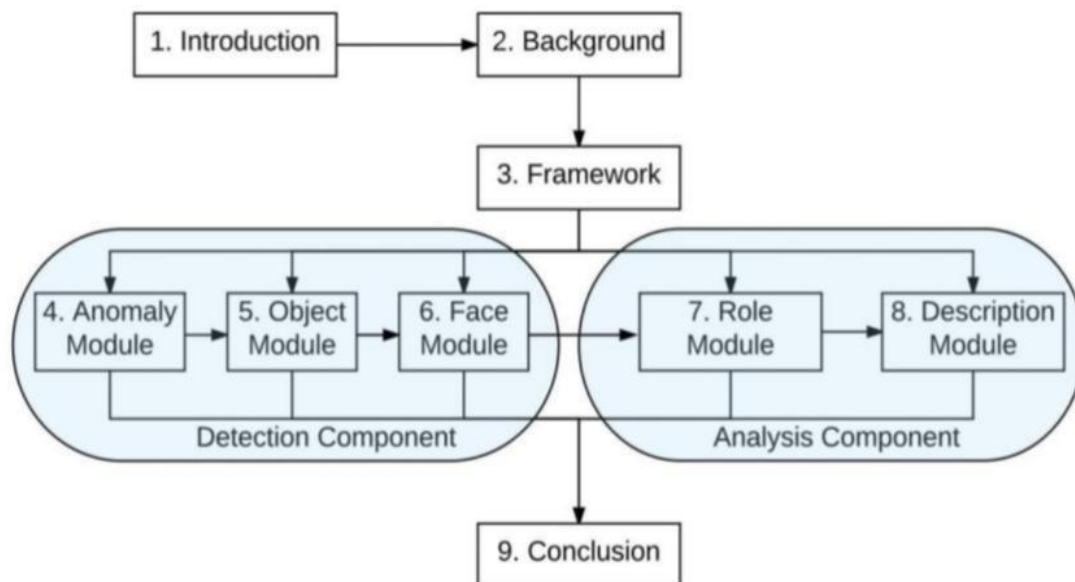


FIGURE 4.1

developed and discussed throughout this thesis. So every method proposed in this thesis can be seen as a system plug-in which could then change the system behaviour.

Note that this thesis does not claim that it has developed a fully functional intelligent surveillance system. It is more reasonable to say that this thesis has made some contributions towards that ultimate goal in the form of exploring novel domains, methods, and research datasets. In the following sections, the relationship between all proposed methods will be described in details. Also, the interaction between the two components will be described

4.1 DESIGN METHODOLOGY:

EXISTING SYSTEM:

Object detection and tracking is one of the critical areas of research due to routine change in motion of object and variation in scene size, occlusions, appearance variations, and ego-motion and illumination changes. Specifically, feature selection is the vital role in object tracking. It is related to many real time applications like vehicle perception, video surveillance and so on. In order to overcome the issue of detection, tracking related to object movement and appearance. Most of the algorithm focuses on the tracking algorithm to smoothen the video sequence. On the other hand, few methods use the prior available information about object shape, color, texture and so on. Tracking algorithm which combines above stated parameters of objects is discussed and analyzed in this research. The goal of this paper is to analyze and review the previous approach towards object tracking and detection using video sequences through different phases.

Also, identify the gap and suggest a new approach to improve the tracking of object over video frame

PROPOSED SYSTEM:

This system has accuracy of about 83 %. Also we use a pi camera. So the image is captured via the pi camera and it is send to the tkinter for processing for face and human detection with the help of openCV. Then, the face detected is

include turning on and turning off the security camera using voice control, record, monitor in / o * u * t operations. Thus, one can provide a low cost security system.

4.2 MODULE DESCRIPTION

The Detection Component

The detection component studies the face detection/alignment/recognition, object detection, anomaly detection functions of video surveillance in both regular domain and novel domains such as Thermal domain, wild animal in uncontrolled nature environment domain, and crowded scene Domain. The details of the detection component will be illustrated below in the form of modules. The process is quite simple in contrast to mainstream as shown in Figure 1.2. We first study the existing methods in novel domains to explore more possibilities of smart video Surveillance. Then based on the observations, experiments and modifications, we propose novel Methods to provide competitive performance while the previous methods generate in novel domains.

Anomaly Module

For a video of a particular time window, the most important task of surveillance will notify the user of strange event/object. For most cases of public surveillance, the scene could be very crowded. To that end, an effort has been made to deal with anomaly detection in crowded scenes.

The brief literature review in section 2.3.4 has introduced the current progress in anomaly detection for the crowded scenario. From there, a novel method is proposed based on trajectory Analysis. Details and experiments will be shown in Chapter 4.

This part of work is making an effort to boost the performance of detection component of current video Surveillance system which, in general, already equipped with anomaly

Object Module

In a detection component, the most important and fundamental problem will be the object detection. In this thesis, DPM is the baseline and foundation for object module and a novel method based on DPM are also proposed for light-weight fast computing purpose. As mentioned in Chapter 1, the speed of the surveillance system is also very important. This part of the work is also an exploration of wild animal domain surveillance in a totally uncontrolled nature environment including daylight/nighttime.

This is a collaboration with the Queensland Government Wildlife and Parks service department who provided the raw data. From there, a refined image dataset is built featuring wild animals such as kangaroos, emus, dingoes and wild cats. The object detection

module developed here can serve for the following analysis component. The provided detection results (bounding boxes) will be saved in the gathered information as shown

in Figure 3.2 which will then be used for the generation of video description in Chapter 8. The DPM-related techniques from this module will also serve as the foundation of face module which will be described below.

Face Module

This part of the work is done during a half-year full-time research project at Canon Information System Research Australia Sydney (Refer as Canon for short in the following context). Due to the company's confidential policy, some work has been protected for patent application and not allowed to be shown in the Thesis.

Apart from that, a significant

amount of work regarding the investigation and experiments on face detection

on alignment and pose normalisation will be shown in Chapter 6. As mentioned above, we study the face

related problem in both the visible light domain and novel thermal domain.

Briefly speaking, extensive face detection and alignment experiments are

component is shown in Figure 3.2. However, they still have connections even if they run in parallel

. For example, with the help of pre-defined rules from an anomaly module, the particular/abnormal object can be detected using the method from object module. The primary purpose of the detection modules is to provide objects/trajectories/activities candidatures for the following analysis component.

Further Discussion on Detection Component

As mentioned before, an important sign of the desired surveillance system is its ability to work in different domains and environments. As part of this thesis's research, thermal domain and night vision wildlife domain are explored as pioneer works. Also, for the first time, extensive data collection and experiments on wild animal surveillance such as kangaroo surveillance are performed in Chapter 1. Note that even the extended modules are connected to the following analysis component. For example, the proposed Kangaroo detector will provide detection bounding box information for video description module in videos containing kangaroos.

THE ANALYSIS COMPOUND

This is the key ingredient which can make the surveillance system interact with human user directly. Being a pioneer work, this thesis aims to show the importance of role classification and its impact on video description. It is unrealistic to build an ideal system that can solve video event analysis perfectly at this stage. The scope of this thesis, as mentioned before, is to contribute to the future surveillance system and spark more interest in this topic. So for now, only role module and description module are researched and integrated as plug-ins for the analysis component.

Role Module

To better understand the content of a video event, it is necessary to determine the relationships between the important objects/animals/persons in this

scene. This is how we understand most social events and perform our daily social activities. This part is called role analysis and it is a novel research area.

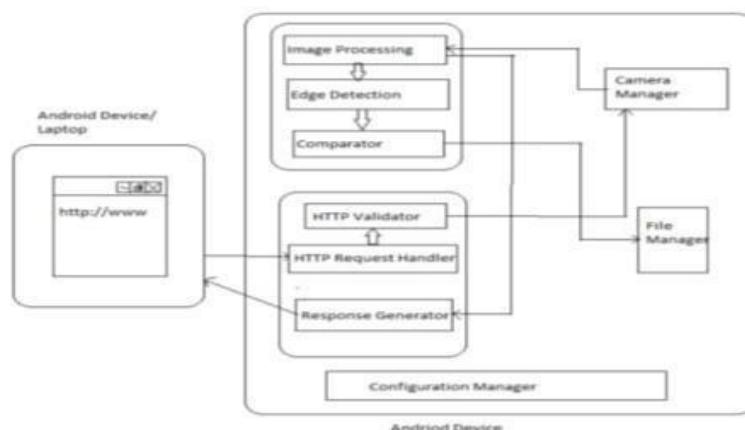
In general, social role analysis is extremely useful for various applications. One could generate a video description once the roles of each detected person are identified. Role analysis is also useful to perform searches in video surveillance. The background of role analysis and proposed novel approach will be described in Chapter 7.

Description Module

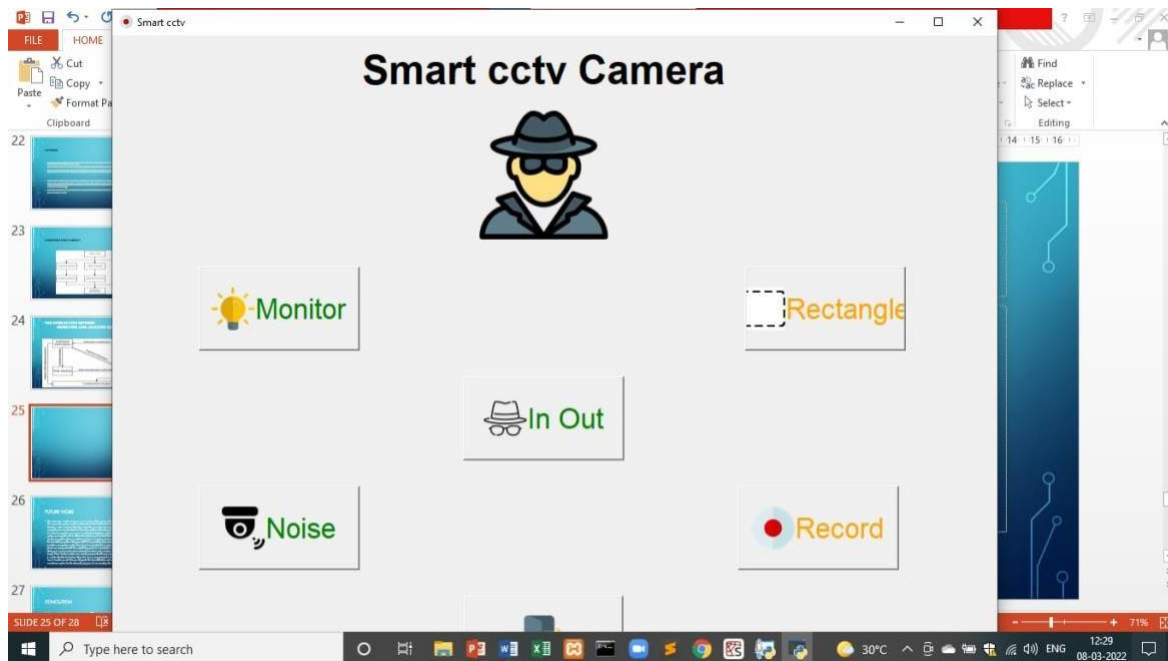
In this stage, the role module serves for the description module. The video description module fuses the output of nearly all previous modules to generate human-understandable natural language description in the form of text. DPM based detectors from Chapter 5 will provide the candidatures for the subject and objects in a simple event of the video. Then the tracking and trajectories refinement techniques from Chapter 4 will help to generate the trajectory based features.

The role classification method from Chapter 7 will come in to assign the roles in an event. Finally, the sentence will be shown to human users with the help of all previous modules. The proposed video description framework is explained in Chapter 8

ARCHITECTURE DIAGRA

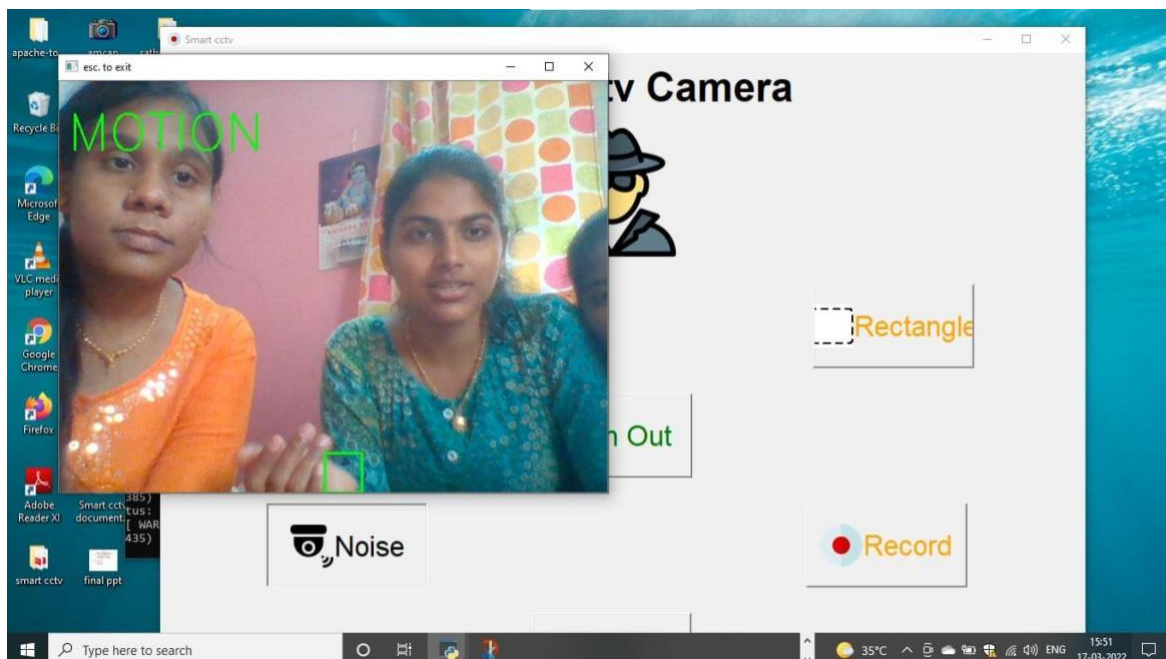


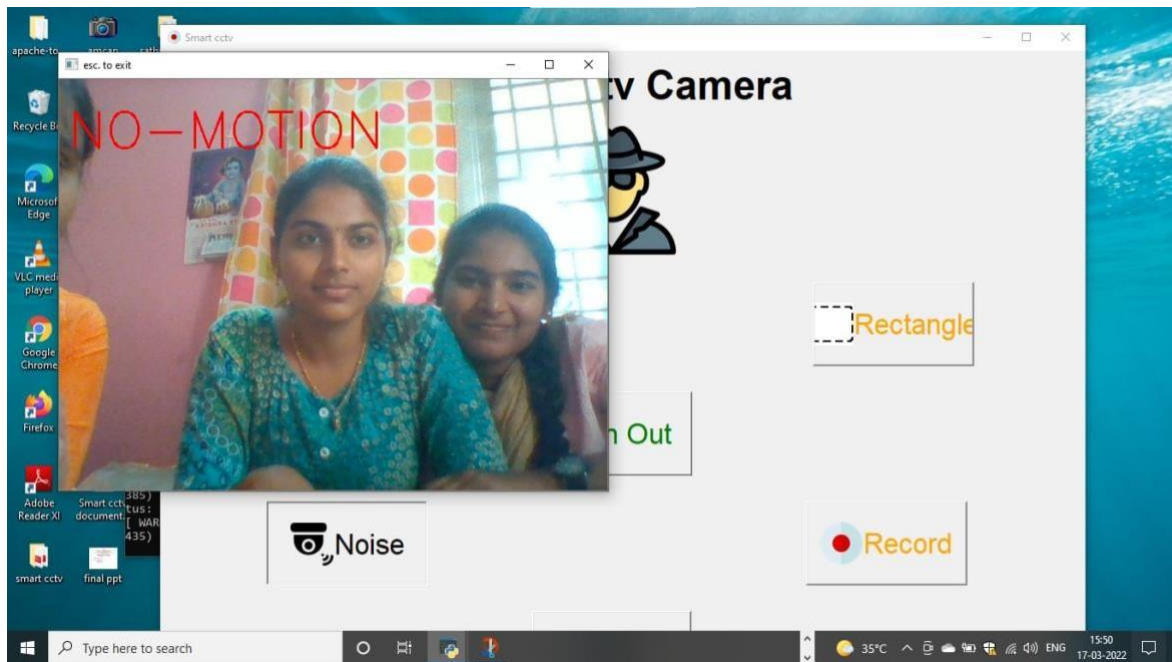
CHAPTER-5- RESULT AND THEORY



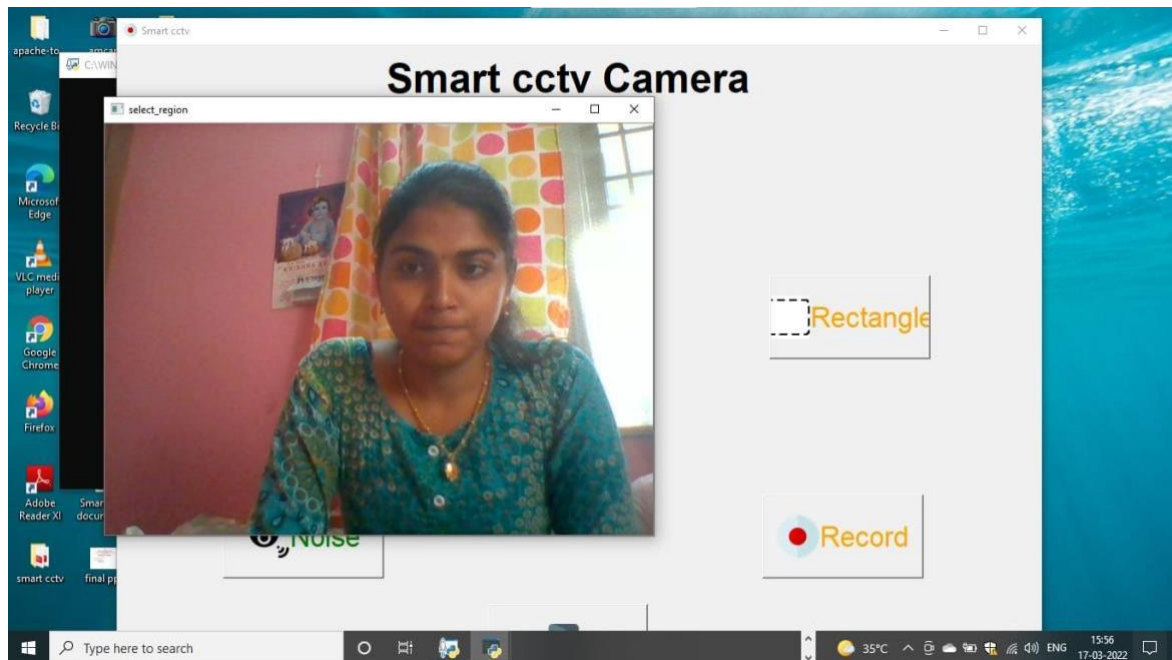
APP FRONT LAYOUT

RECT NOICE

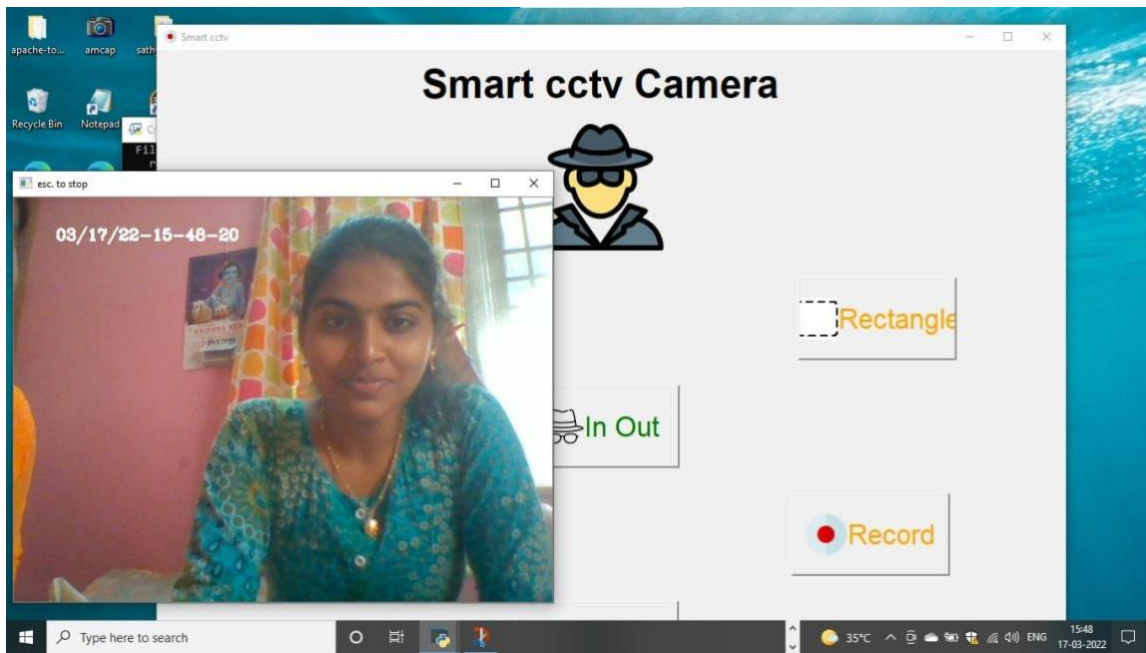




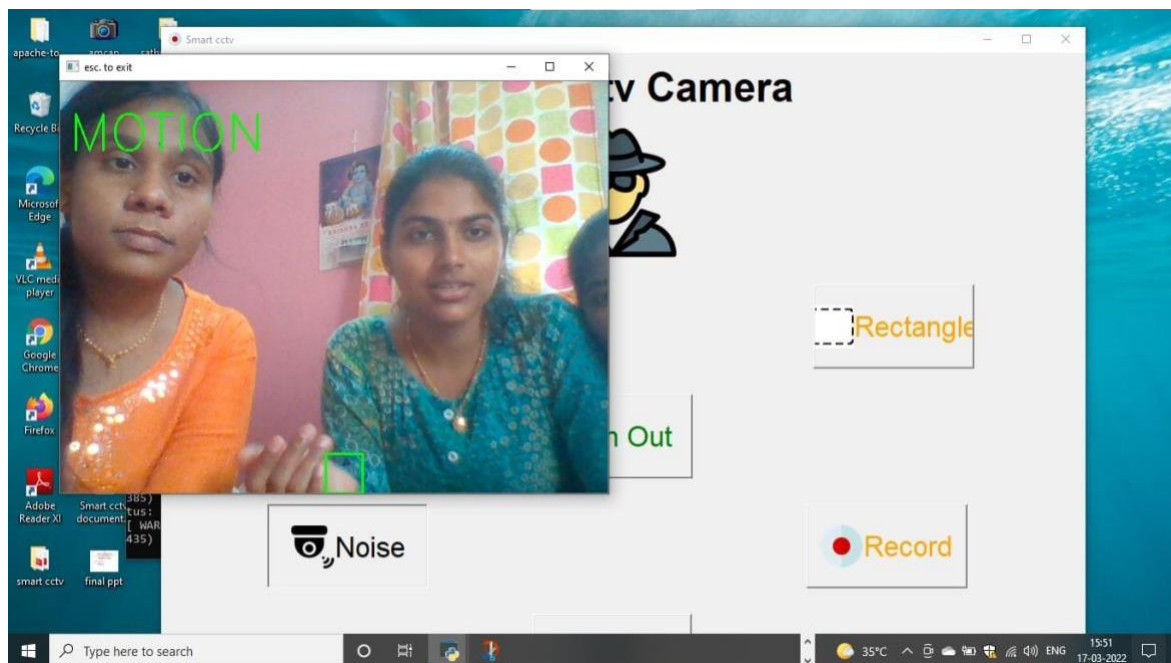
JUST FOR TEST



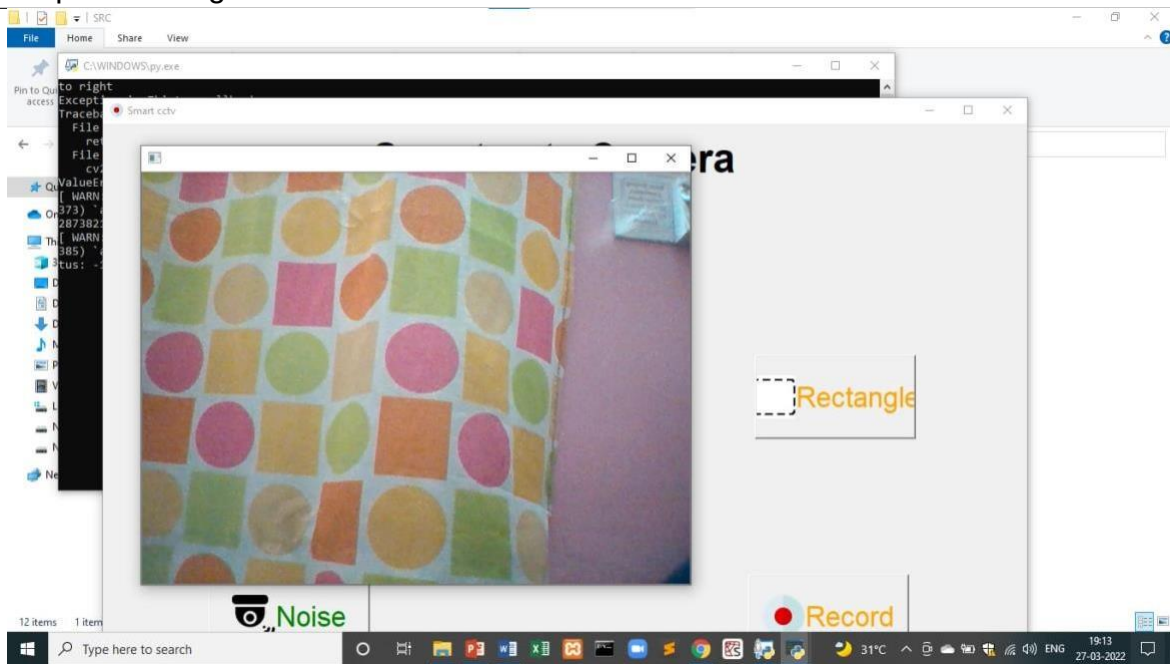
RECORD



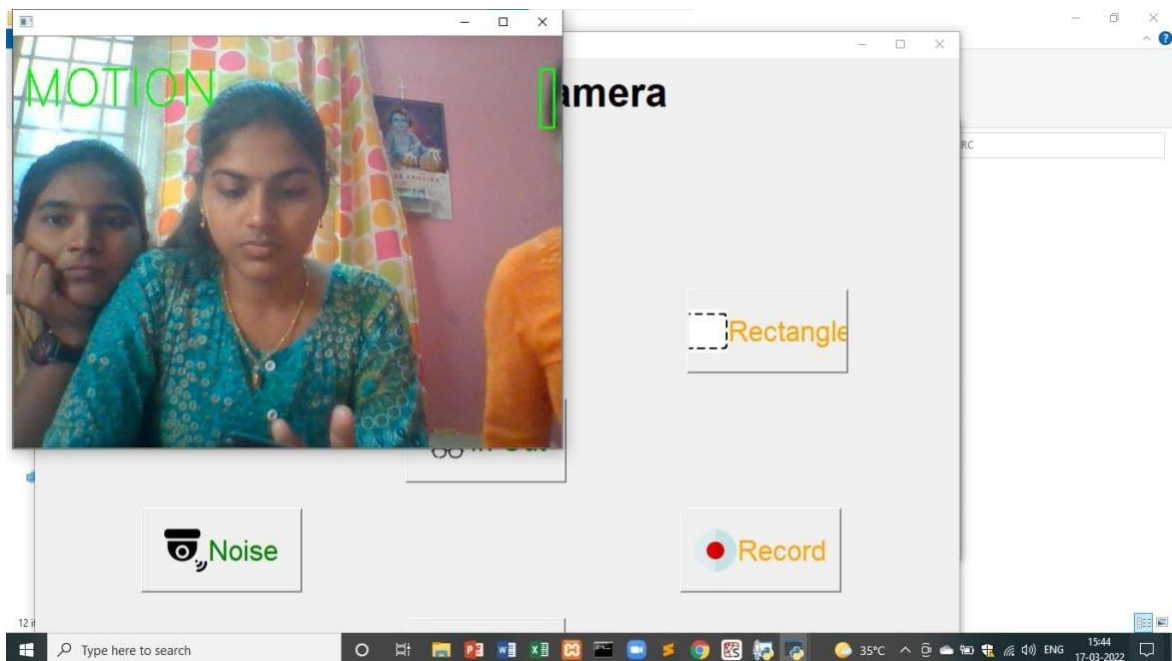
IN AND OUT



Chapter2.BackgroundandLiteratureRe



RECTANGLE



6:CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION:

Thus, we have developed a smart CCTV camera that can be started using and is capable of providing face detection a. Also the camera system is compact and can be implemented with low cost. The implemented face detection algorithm (Haar like cascade classifier) is very effective, with an accuracy of 88.9 percent which can be increased further by effectively improving the illumination of the area. However, this system is connected with the help of OPENCV. This can be overcome by making the system wireless.

FUTURE WORK:

This system has a wide range of uses in various fields, such as banking, forensic department, etc The reason this system is quiet useful is due to the fact that it is highly compact and it provides face detection and an instant notification about the same through email. In addition to this face recognition can also be tried in future. Recognition is the main part of any security system. Usually for a best recognition system, we require a well-trained database, which can provide the base for our recognition. So to obtain the database, first collect the images of the subject individual for the recognition. Once we obtain and train our system, we can provide face recognition. We use the local binary pattern histogram (LBPH) for providing face recognition. This method helps us to provide a recognition model. The image is converted into a gray scale image. Then, the image pixels are compared with the neighboring pixels in a clock-wise or anti-clock-wise manner. Histogram is performed and normalization is done and a feature vector is generated for every image. These feature vectors can now be processed with some algorithms to classify images which is used to identify the texture. Once the face is recognized, it is checked to see if the detected face is familiar or not. Thus we integrate

the face detection and recognition to provide a smart surveillance system for the domestic purposes in our everyday life.

REFERENCE

- [1] "You're being watched, New York!" in BBC, March 2002.
- [2] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, vol. 30, no. 3, pp. 555–560, 2008.
- [3] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.
- [4] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual Question Answering," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 2425–2433.
- [5] A. Athanasios, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *Computer Vision and Pattern Recognition (CVPR)*, *IEEE Conference on*, June 2013, pp. 3444–3451.
- [6] D. Ballard, "Animate vision," *Artificial Intelligence*, vol. 48, pp. 57–86, 1991.
- [7] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *CVPR*, 2012, pp. 1830–1837.
- [8] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [9] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 930–940, Dec 2013.

- [10] J. Bouguet, "Pyramidal implementation of the Lucas-Kanade feature tracker: description of the algorithm," in *Microprocessor Research Lab, Intel Corporation*, 1999.
- [11] M. Brezovan and C. Badica, "A review on vision surveillance techniques in smart home environments," in *2013 19th International Conference on Control Systems and Computer Science*, May 2013, pp. 471–478.
- [12] R. Brooks, R. Greiner, and T. Binford, "The acronym model-based vision system," in *6th Int. Joint Conf. on Artificial Intelligence*, 1979.
- [13] T. Burghardt and J. Calic, "Real-time face detection and tracking of animals," in *2006 8th Seminar on Neural Network Applications in Electrical Engineering*, Sept 2006, pp. 27–32.
- [14] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.
- [15] Y. Chang, C. Hu, and M. Turk, "Manifold of facial expression," in *Analysis and Modeling of Faces and Gestures, IEEE International Workshop on*, Oct 2003, pp. 28–35.
- [16] C. Chen and K. Grauman, "Efficient activity detection with max-subgraph search," in *Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1274–1281.
- [17] G. Chen, T. X. Han, Z. He, R. Kays, and T. Forrester, "Deep convolutional neural network based species recognition for wild animal monitoring," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 858–862.
- [18] Q. Chen, J. Huang, R. Feris, L. M. Brown, J. Dong, and S. Yan, "Deep domain adaptation for describing people based on fine-grained clothing attributes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 5315–5324.
- [19] T. Clutton-Brock and B. Sheldon, "Individuals and populations: the role of long-term, individual-based studies of animals in ecology and evolutionary biology," *Trends in Ecology and Evolution*, vol. 25, pp. 562–573, Sept. 2010.

APPENDIX SOURECE

CODE1.MAIN:

```
Import tkinter as tk Import tkinter.font
as fontFrom in_out import in_outFrom
motion import noise
From rect_noise import rect_noiseFrom record
import record
From PIL import Image, ImageTk

Window = tk.Tk()
Window.title("Smart cctv")
Window.iconphoto(False, tk.PhotoImage(file='mn.png'))
Window.geometry('1080x760')

Frame1 = tk.Frame(window)

Label_title = tk.Label(frame1, text="Smart cctv Camera")
Label_font = font.Font(size=35, weight='bold',family='Helvetica')
Label_title['font'] = label_font
Label_title.grid(pady=(10,10), column=2)
```

```
Icon = Image.open('icons/spy.png')  
Icon = icon.resize((150,150),  
Image.ANTIALIAS)Icon =  
ImageTk.PhotoImage(icon)  
Label_icon = tk.Label(frame1, image=icon)  
Label_icon.grid(row=1, pady=(5,10), column=2)
```

```
Btn1_image = Image.open('icons/lamp.png')  
Btn1_image = btn1_image.resize((50,50),  
Image.ANTIALIAS)Btn1_image =  
ImageTk.PhotoImage(btn1_image)
```

```
Btn2_image = Image.open('icons/rectangle-of-cuttetd-line-geometrical-shape.png')  
Btn2_image = btn2_image.resize((50,50),  
Image.ANTIALIAS)Btn2_image =  
ImageTk.PhotoImage(btn2_image)
```

```
Btn5_image = Image.open('icons/exit.png')  
Btn5_image = btn5_image.resize((50,50),  
Image.ANTIALIAS)Btn5_image =  
ImageTk.PhotoImage(btn5_image)
```

```
Btn3_image = Image.open('icons/security-camera.png')  
Btn3_image = btn3_image.resize((50,50),  
Image.ANTIALIAS)Btn3_image =  
ImageTk.PhotoImage(btn3_image)
```

```
Btn6_image = Image.open('icons/incognito.png')
```

```
Btn6_image = btn6_image.resize((50,50),
```

```
Image.ANTIALIAS)Btn6_image =
```

```
ImageTk.PhotoImage(btn6_image)
```

```
Btn4_image = Image.open('icons/recording.png')
```

```
Btn4_image = btn4_image.resize((50,50),
```

```
Image.ANTIALIAS)Btn4_image =
```

```
ImageTk.PhotoImage(btn4_image)
```

```
#.....Button
```

```
.....#Btn_font =
```

```
font.Font(size=25)
```

```
Btn1 = tk.Button(frame1, text='Monitor', height=90, width=180,  
fg='green',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=4, 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)
```

```
Frame1.pack()
```

```
Window.mainloop
```

```
()
```

2.IN_OUT:

```
Import cv2
```

From datetime import

datetimeDef in_out():

Cap = cv2.VideoCapture(0)

Right, left = "", ""

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)
```

```
If right == "" and left == "":
```

```
    If x > 500:
```

```
        Right = True
```

```
    Elif x < 200:
```

```
        Left =
```

```
        True
```

```
    Elif right:
```

```
        If x < 200:
```

```
            Print("to left")
```

```
            X = 300
```

```
            Right, left = "", ""
```

```
            Cv2.imwrite(f'visitors/in/{datetime.now().strftime('%-y-%m-%d-%H:%M:%S')}.jpg', frame1)
```

Elif left:

 If x > 500:

 Print("to right")

 X = 300

 Right, left = "", ""

 Cv2.imwrite(f"visitors/out/{datetime.now().strftime('%y-%m-%d-%H:%M:%S')}.jpg",
 frame1)

Cv2.imshow("", frame1)

K = cv2.waitKey(1)

 If k == 27:

 Cap.release()

 Cv2.destroyAllWindows()

 Break

this is change

made#one more

3.JUST_FOR_TEST:

```
Import cv2
```

```
Cap = cv2.VideoCapture(0)
```

```
Fourcc = cv2.VideoWriter_fourcc(*'XVID')
```

```
Out = cv2.VideoWriter('output.avi', fourcc,20.0,(640,480))
```

```
While True:
```

```
    _, frame = cap.read()
```

```
    Out.write(frame)
```

```
    Cv2.imshow("", frame)
```

```
    If cv2.waitKey(1) == 27:
```

```
        Cap.release()
```

```
        Cv2.destroyAllWindows()
```

```
        Break
```

4.MOTION:

```
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
```

5.RECORD:

```
import cv2
```

```
from datetime import datetime
```

```
def record():
```

```
    cap = cv2.VideoCapture(0)
```

```
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
```

```
    out = cv2.VideoWriter(f'recordings/{datetime.now().strftime("%H-%M-%S")}.avi', fourcc, 20.0, (640, 480))
```

```
    while True:
```

```
        _, frame = cap.read()
```

```
cv2.putText(frame, f'{datetime.now().strftime("%D-%H-%M-%S")}',  
(50,50),cv2.FONT_HERSHEY_COMPLEX,  
0.6, (255,255,255), 2)
```

```
out.write(frame)
```

```
cv2.imshow("esc. to stop", frame)
```

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

6.RECT_NOICE:

```
import cv2
```

```
done1 = False
```

```
doner = False
```

```
x1,y1,x2,y2 = 0,0,0,0
```

```
def select(event, x, y, flag,
          param):global
    x1,x2,y1,y2,donel, doner
    if event ==
        cv2.EVENT_LBUTTONDOWN:
            x1,y1 = x,y
            donel = True
    elif event ==
        cv2.EVENT_RBUTTONDOWN:
            x2,y2 = x,y
            doner = True
            print(doner,
                donel)

    def
    rect_noise():

    global x1,x2,y1,y2, donel,
    donercap =
    cv2.VideoCapture(0)

    cv2.namedWindow("select_region")
    cv2.setMouseCallback("select_region", select)

    while True:
        _, frame = cap.read()
```

```
cv2.imshow("select_region", frame)
```

```
if cv2.waitKey(1) == 27 or doner == True:
```

```
    cv2.destroyAllWindows()
```

```
    print("gone--
```

```
    ")break
```

```
while True:
```

```
    _, frame1 = cap.read()
```

```
    _, frame2 = cap.read()
```

```
    frame1only = frame1[y1:y2,
```

```
    x1:x2]frame2only =
```

```
    frame2[y1:y2, x1:x2]
```

```
    diff = cv2.absdiff(frame2only, frame1only)
```

```
    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+x1, y+y1), (x+w+x1, y+h+y1), (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.rectangle(frame1, (x1,y1), (x2, y2), (0,0,255), 1)  
cv2.imshow("esc. to exit", frame1)  
  
if cv2.waitKey(1) == 27: cap.release()  
    cv2.destroyAllWindows()break
```