SMART-CCTV

19ITPN6601
INNOVATIVE AND CREATIVE PROJECT REPORT

Submitted by

LOGITH N (19BIT038)
KAMESHWARAN M (19BIT056)
JEEVA S (19BIT064)

in partial fulfillment for the award of the degree

of

Bachelor of Technology

in

Information Technology

Dr. Mahalingam College of Engineering and Technology

Pollachi - 642003

An Autonomous Institution

Affiliated to Anna University, Chennai - 600 025

JUNE 2022

Dr. Mahalingam College of Engineering and Technology

Pollachi - 642003

An Autonomous Institution

Affiliated to Anna University, Chennai - 600 025

BONAFIDE CERTIFICATE

Certified that this Innovative and Creative Project Report, **SMART-CCTV** is the bonafide work of

LOGITH N(19BIT038)
KAMESHWARAN M(19BIT056)
JEEVA S(19BIT064)

who carried out the project work under my supervision.

_____

Dr. S. PONNI@SATHYA
Project Guide
Assistant Professor (SG)
Information Technology
Dr. Mahalingam College of Engineering and Technology, NPT-MCET Campus
Pollachi – 642003 India

_____

Dr. S. RAMAKRISHNAN
Professor & Head
Information Technology
Dr. Mahalingam College of Engineering and Technology,
NPT-MCET Campus
Pollachi – 642003 India

Submitted for the Autonomous End Semester Innovative and Creative Project Examination held on _____

INTERNAL EXAMINER 1 **INTERNAL EXAMINER 2**

SMART-CCTV

ABSTRACT

SMART-CCTV is the use of automatic video analysis technologies in video surveillance applications. Since, we are using Computer vision in this project there is no need for anyone watch the boring surveillance videos for 24/7.Using this project, we can reduce Human work, Time, Money which is more needed in the surveillance field. Our Project aims at providing one such idea to ensure safety and security of one's own property. Smart-CCTV can perform Monitoring (which records video when face is detected), Authorize (Records video when unknown person is identified), Identify (Identifies the missing object), Record (Records video like a normal CCTV). Our security systems involve the use of web camera for capturing the frames which then used for video processing functions mentioned above. In our project we had added more quality features. We can easily detect intrusion without seeing the whole recording. Our project plays a major role in saving time and storage. In our project we had used python language alone so it will be a big advantage to carry and implement many things in our application.

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of this project depends largely on the encouragement and guidelines of many others. We take this opportunity to praise the almighty and express our gratitude to the people who have been instrumental in the successful completion of our project.

We wish to acknowledge with thanks for excellent encouragement given by the management of our college and we thank **Dr. C. Ramaswamy, M.E., Ph.D., FIV,** Secretary, NIA Educational Institutions for providing us with a plethora of facilities in the campus to complete our project successfully.

We wish to express our hearty thank to **Dr. A. Rathinavelu, M.Tech., Ph.D.,** Principal of our college, for his constant motivation regarding our project work.

We express our extreme gratefulness to **Dr. S. Ramakrishnan, M.E., Ph.D.,** Professor & Head of the Department, Information Technology for his instrumental in providing such wonderful support to us.

It is also our primary duty to thank our Project Co-ordinators **Dr.J.Ramprasath, M.E., Ph.D.,** Assistant Professor (SS), Information Technology is the backbone of all our project activities, for their consistent guidance and encouragement, which kept us fast and pro-active in our work. It is their enthusiasm and patience that guided us through the right path.

We would like to show our greatest appreciation to **Dr.S.Ponni@Sathya, M.E., Ph.D.,** Assistant Professor (SG), Information Technology. We can't say thank you enough for his tremendous support and help. We feel motivated and encouraged every time we attend his meeting. Without his encouragement and guidance our project would not have materialized.

Finally, we extend our heartfelt thanks to our parents and friends for their constant support throughout this project. The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

OpenCV	–	Open-source Computer Vision library
AI	–	Artificial Intelligence
ML	–	Machine Learning

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Objective of Project

Smart CCTV is a Python GUI application which can run on windows, uses webcam and Computer vision to provide surveillance ability to your laptop/pc and has number of features which are not in normal CCTV which can reduce numerous resources used in Surveillance Field.

1.2 Existing System

We have some of the existing systems for SMART-CCTV in different criteria.

Normal CCTV:

- Where there is always need for a person for 24 hours monitoring
- Can only perform few functions
- High cost
- High Maintenance

1.3 Proposed System

Smart-CCTV can perform number of features which are not in normal CCTV. By using our project, The need of man power and high volume storage disk can be neglected since our project is automated and records only when needed. It performs four functions they are

Namely:

- Monitoring
- Identify the Missing the object
- Authorization
- Recording

Monitoring:

It is used to monitor the surveillance in an entry restricted area.

Identify the Missing object:

It is used to identify the missing object when the motion is stopped once and the differentiation will be shown afterwards.

Authorization:

It is used to record video when unknown person is identified. The image of the authorized person is stored in the database.

Recording:

It is to record the surveillance like a normal CCTV.

1.4 Software Requirement Specification:

Windows/Linux/Mac OS any version, hence it can run on any platform. Python3, it need python to be installed in your system to run this successfully. Packages in python to be installed are OpenCV, skimage, NumPy, tkinter.

CHAPTER 2

DESIGN

2.1 Introduction:

Smart CCTV is a Python GUI application which can run on windows, uses webcam and Computer vision to provide surveillance ability to your laptop/pc and has number of features which are not in normal CCTV which can reduce numerous resources used in Surveillance Field.

2.2 Waterfall Model:

Classical waterfall model is the basic software development life cycle model. It is very simple but idealistic. Earlier this model was very popular but nowadays it is not used. But it is very important because all the other software development life cycle models are based on the classical water model.

Classical waterfall model divides the life cycle into a set of phases. This model considers that one phase can be started after completion of the previous phase. That is the output of one phase will be the input to the next phase. Thus, the development process can be considered as a sequential flow in the waterfall.

Here the phases do not overlap with each other. The different sequential phases of

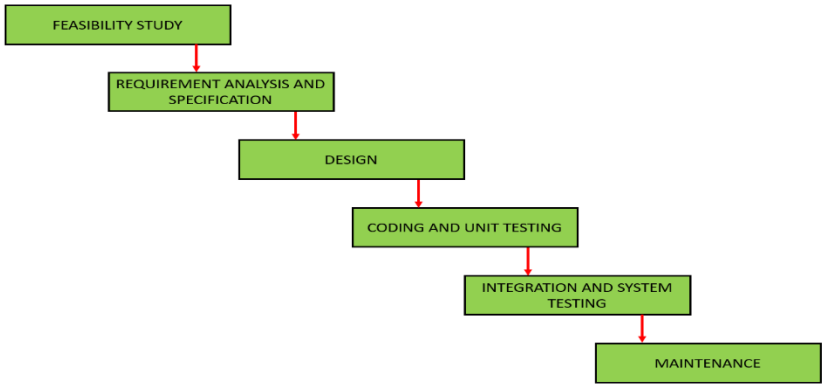

Figure 1.1 Waterfall Model

2.2 UML Diagram:

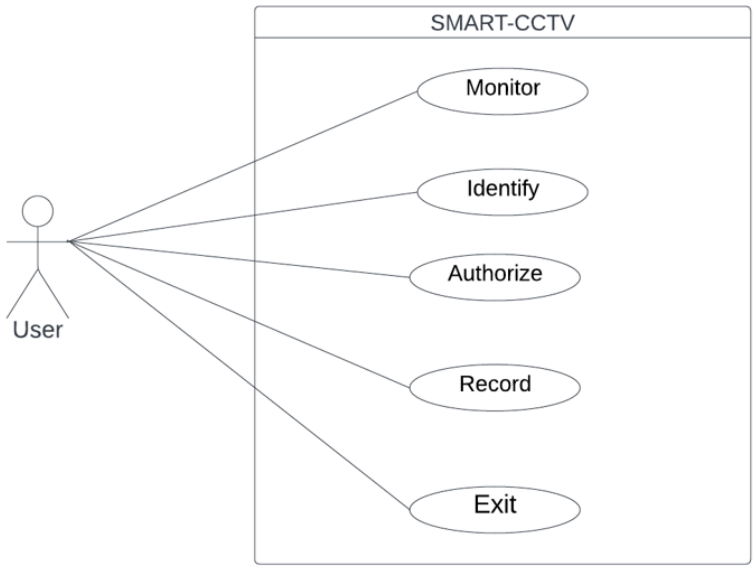

Figure 2.1: Use Case Diagram

The Figure 2.1 shows that Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. In our system customer is an actor either existing user or new user both can interact with our system at the level of ease. On the other side of our system there are two actors who also interact with the main thing of the system where one is the controller of complete system and other is one who gets benefits and obligations through our system.

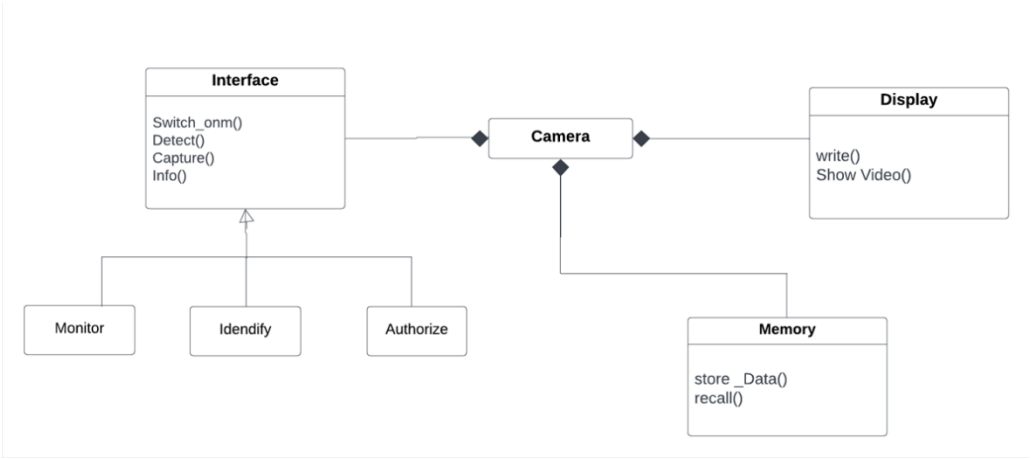

Figure 2.2: Class Diagram

The Figure 2.2 shows that class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

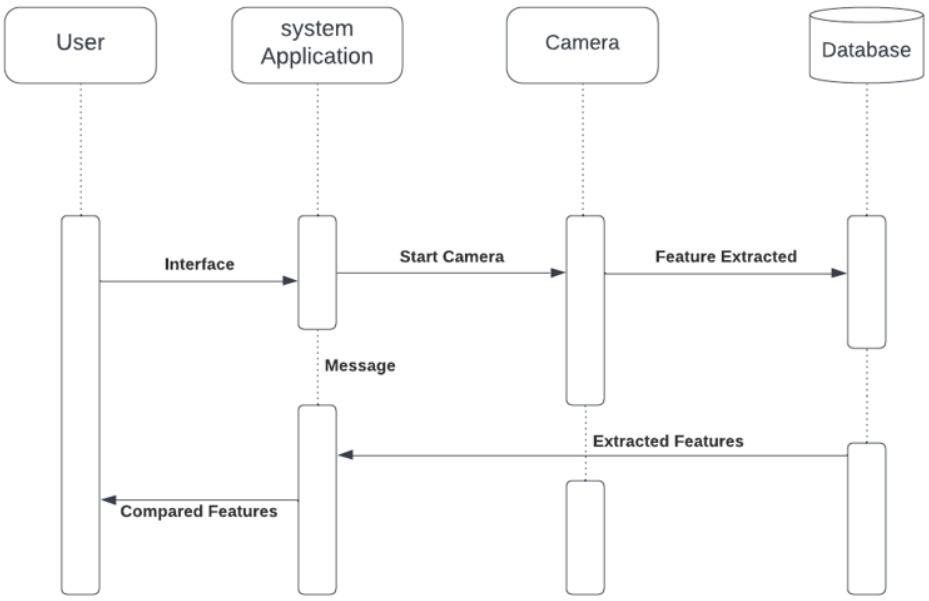

Figure 2.3: Sequence Diagram

The Figure 2.3 shows that sequence diagram or system sequence diagram (SSD) shows object interactions arranged in time sequence in the field of software engineering. This diagram shows the interaction between the objects of Course, Student, Profile, Marks and Batch. All of these pages can be accessed after successful login. The Diagram below helps demonstrates how the login page works in our project.

.

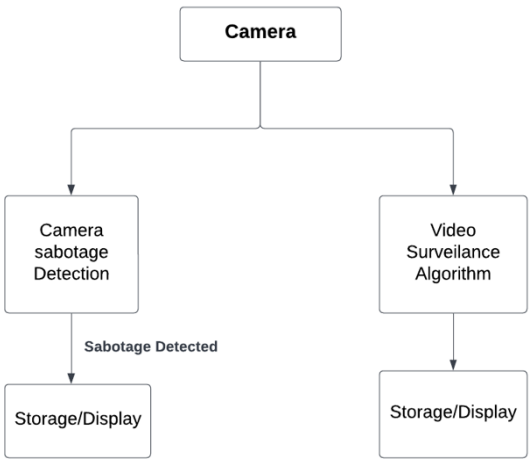

Figure 2.4: Communication Diagram

The Figure 2.4 shows that Communication diagrams are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, communication diagrams are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case. As explained above in the sequence diagram all interactions are based on client-server model.

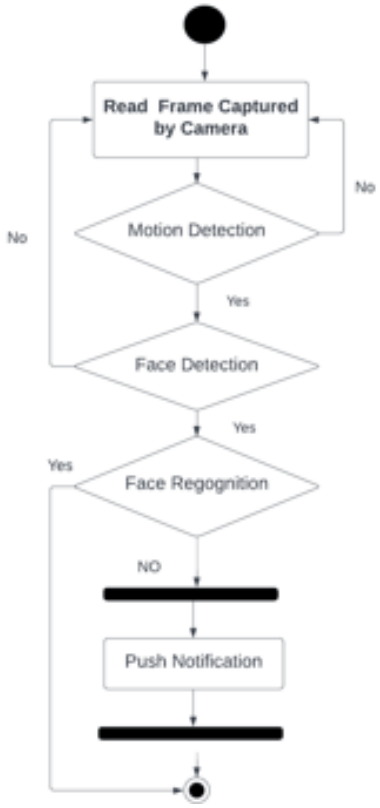

Figure 2.5: Activity Diagram

The Figure 2.5 shows that an activity diagram shows business and software processes as a progression of actions. These actions can be carried out by people, software components or computers. Activity diagrams are used to describe business processes and use cases as well as to document the implementation of system processes. The main process of accessing the system is given in the activity diagram and thus every process at the starting is tend to fork node where user have various options of using the website and thus finally after performing their desired action everything is tend to merge and get out of it from the final node.

Figure 2.6: State Chart Diagram

The Figure 2.6 shows that a state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams. In the above state chart diagram, each and every event triggered by both of the user and the admin is drawn. Firstly, admin and staff sign in moves to the join to merge from initial value. After that the credentials are verified using database. After verification Both reaches managing record module. After that it was splitted up. Then it reaches assigning and profile maintenance. Then it gathers details and gets logged out.

CHAPTER 3

IMPLEMENTATION & RESULTS

3.1 Introduction:

Smart CCTV is a Python GUI application which uses opencv, skimage and face-recognition in-built Computer vision libraries to perform image and video processing. Since, we are using Computer vision in this project there is no need for anyone watch the surveillance videos for 24/7. Using this project, we can reduce human work**,** time, storage devices.

3.2 Module Description:

Monitoring:

It is used to monitor the surveillance in an entry restricted area.

Identify the Missing object:

It is used to identify the missing object when the motion is stopped once and the differentiation will be shown afterwards.

Authorization:

It is used to record video when unknown person is identified. The image of the authorized person is recorded in the database.

Recording:

It is to record the surveillance like a normal CCTV.

3.4 Result:

We have successfully implemented all four modules Monitor, Identify, Authorize, Record and by developing SMART-CCTV.

CHAPTER 4

CONCLUSION & FUTURE WORK

4.1 Project Conclusion:

The project is as much completed and implemented successfully. Here we can use the app to watch or in other words it gives vision capability to computers like meets the objectives of theft reduction, asset protection, security investigations, providing evidence and deterrence.

4.2 Future Enhancement:

Below are some future works are going to be added on this project:

- Adding in-built night vision capability.
- More feature such as
- Deadly weapon detection
- Accident detection
- Fire Detection

APPENDIX I: SCREEN SHOTS:

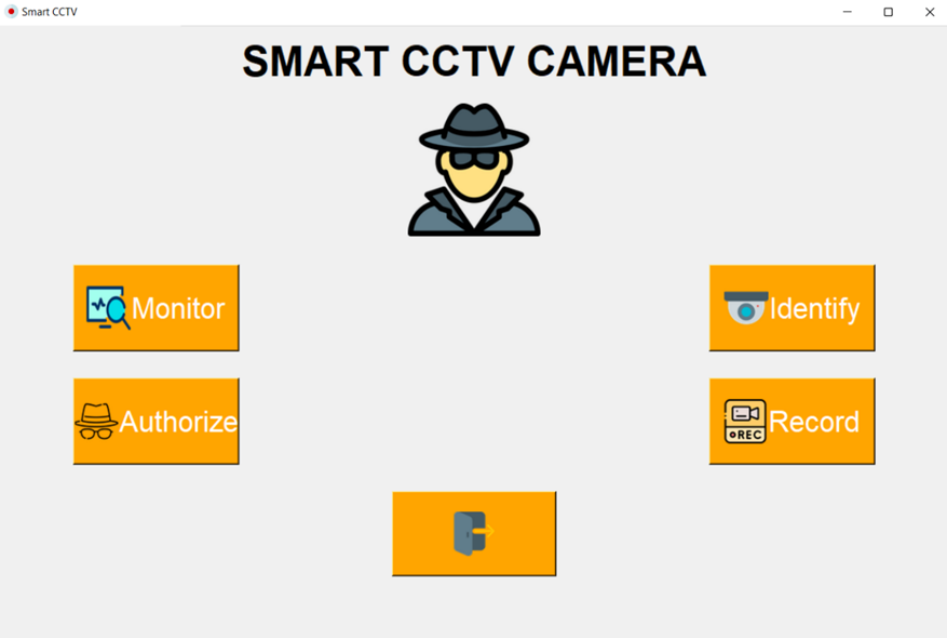

Figure 3.1 Home page

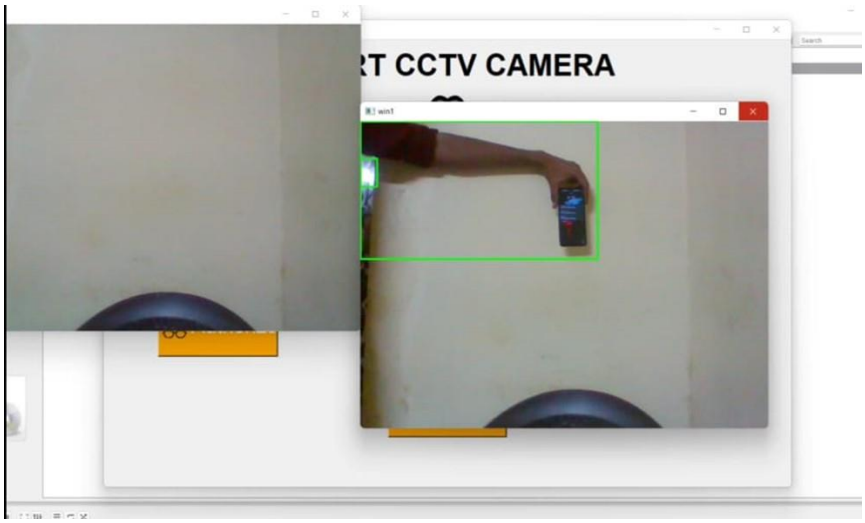

Figure 3.2: Identify

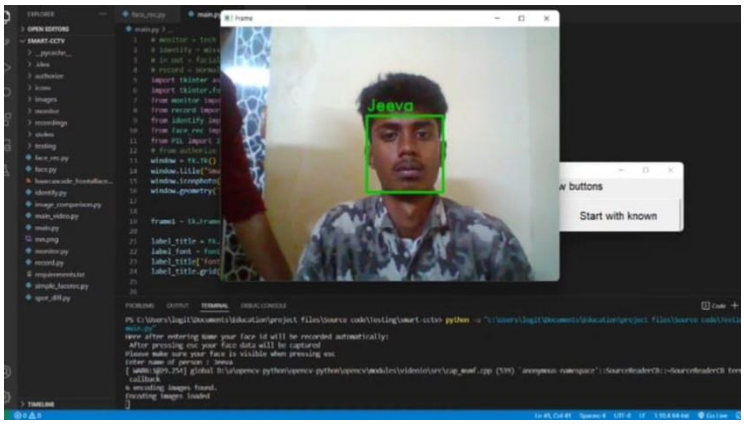

Figure 3.3: Authorize

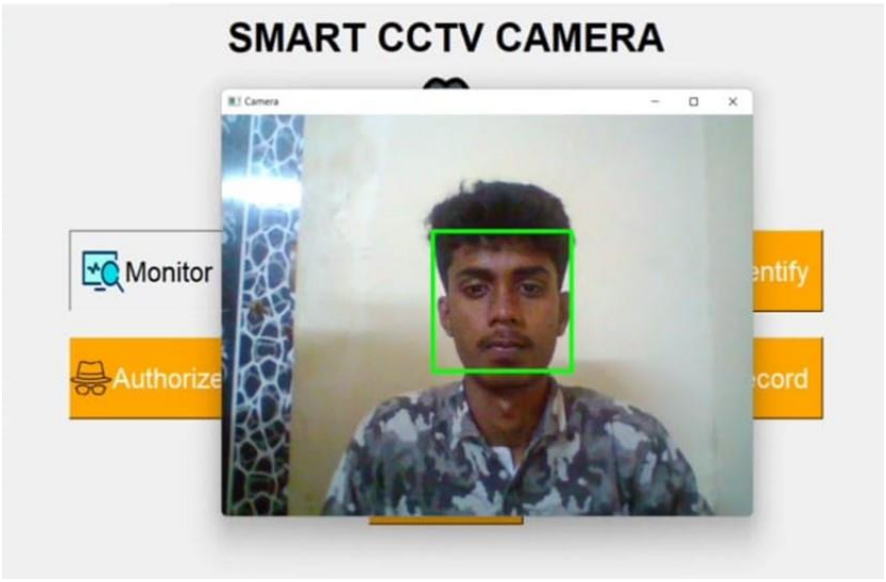

Figure 3.4: Monitor

Figure 3.5: Recording

SAMPLE CODE:

Home page:

```
import tkinter as tk
import tkinter.font as font
from monitor import monitor
from record import record
from identify import identify
from face_rec import maincall
from PIL import Image, ImageTk
window = tk.Tk()
window.title("Smart CCTV")
window.iconphoto(False, tk.PhotoImage(file='mn.png'))
window.geometry('1080x700')


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.Resampling.LANCZOS)
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/Monitor.png')
btn1_image = btn1_image.resize((50, 50), Image.Resampling.LANCZOS)
btn1_image = ImageTk.PhotoImage(btn1_image)

btn5_image = Image.open('icons/exit.png')
btn5_image = btn5_image.resize((50, 50), Image.Resampling.LANCZOS)
btn5_image = ImageTk.PhotoImage(btn5_image)

btn6_image = Image.open('icons/incognito.png')
btn6_image = btn6_image.resize((50, 50), Image.Resampling.LANCZOS)
btn6_image = ImageTk.PhotoImage(btn6_image)

btn4_image = Image.open('icons/rec.png')
btn4_image = btn4_image.resize((50, 50), Image.Resampling.LANCZOS)
btn4_image = ImageTk.PhotoImage(btn4_image)

btn7_image = Image.open('icons/main.png')
```

```
btn7_image = btn7_image.resize((50, 50), Image.Resampling.LANCZOS)
btn7_image = ImageTk.PhotoImage(btn7_image)


# Button
# Monito Button
btn_font = font.Font(size=25)
btn1 = tk.Button(frame1, text='Monitor', height=90, width=180, fg='white',
command=monitor, image=btn1_image, compound='left')
btn1['font'] = btn_font
# btn1["border"] = "0"
btn1["bg"] = "orange"
btn1.grid(row=3, pady=(20, 10))
# Record Button
btn4 = tk.Button(frame1, text='Record', height=90, width=180, fg='white',
command=record, image=btn4_image, compound='left')
btn4['font'] = btn_font
btn4["bg"] = "orange"
btn4.grid(row=5, pady=(20, 10), column=3)
# Authorization
btn6 = tk.Button(frame1, text='Authorize', height=90, width=180, fg='white',
command=maincall, image=btn6_image, compound='left')
btn6['font'] = btn_font
btn6["bg"] = "orange"
btn6.grid(row=5, pady=(20, 10))
# Exit
btn5 = tk.Button(frame1, height=90, width=180, fg='red', command=window.quit,
image=btn5_image)
btn5['font'] = btn_font
btn5["bg"] = "orange"
btn5.grid(row=6, pady=(20, 10), column=2)
# Identify Button
btn7 = tk.Button(frame1, text="Identify", fg="white", command=identify,
compound='left', image=btn7_image, height=90, width=180)
btn7['font'] = btn_font
btn7["bg"] = "orange"
btn7.grid(row=3, column=3, pady=(20, 10))

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

Recording:

```
import cv2
import datetime


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

fourcc = cv2.VideoWriter_fourcc(*'mp4v')
# out = cv2.VideoWriter(f'recordings/{datetime.now().strftime("%H-%M-%S")}.avi',
fourcc,20.0,(640,480))
frame_size = (int(cap.get(3)), int(cap.get(4)))

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

current_time = datetime.datetime.now().strftime("%d-%m-%Y-%H-%M-%S")
out = cv2.VideoWriter(f"recordings/{current_time}.mp4", fourcc, 20, frame_size)
out.write(frame)
cv2.imshow("esc. to stop", frame)

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

Spot Difference:

```
import cv2
import time
import skimage
from skimage.metrics import structural_similarity
from datetime import datetime
import beepy


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", frame2)
    cv2.imshow("win1", frame1)
    beepy.beep(sound=4)
    cv2.imwrite("stolen/" + datetime.now().strftime('%d-%m-%Y-%H-%M-%S') +
".jpg", frame1)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
   return 1
```

WEB REFERENCE :

https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/

https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e

https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9

https://opencv.org/