**SMART CCTV**

**Minor Project Report**

*Submitted in partial fulfilment of the requirement of the degree of*

**BACHELOR OF TECHNOLOGY**

*to*

**K.R Mangalam University**

*by*

**Student Name (Roll)**

**Student Name (Roll)**

**Student Name (Roll)**

Under the supervision of

**Supervisor Name Supervisor Name**

**<Internal> <External>**

**Designation Designation**

**Company Name**



Department of Computer Science and Engineering

School of Engineering and Technology

K.R Mangalam University, Gurugram- 122001, India

April 2025

**CERTIFICATE**

This is to certify that the Project Synopsis entitled, “**SMART CCTV**” submitted by “**Bhavya Gauri(19CSU066), Harsh Yadav(19CSU) and Hridyesh(19CSU)”** to **K.R Mangalam University, Gurugram, India,** is a record of bonafide project work carried out by them under my supervision and guidance and is worthy of consideration for the partial fulfilment of the degree of **Bachelor of Technology** in **Computer Science and Engineering** of the University.

**Type of Project (Tick One Option)**

**Industry/Research/University Problem**

<Signature of Internal supervisor>  
<Name and designation of supervisor>

Signature Dean SOET

Dr Pankaj Agarwal, Dean SOET

Date: 3rd April 2025

**Project Use Case & Deployment Confirmation Certificate**

This is to formally acknowledge that the following student(s) from K.R. Mangalam University have successfully undertaken and completed an industry-based project under our mentorship, in alignment with the stated objectives and requirements of our organization.

**Project Details:**

* **Project Title:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* **Domain/Technology Used:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* **Industry Use Case / Business Problem Addressed:**
* **Expected Outcome/Utility of the Project in Our Organization:**

**Student Details:**

| **Name of Student** | **Enrollment No.** | **Program** | **Year** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

**Mentor Declaration & Disclaimer:**

I, the undersigned, hereby declare that:

1. The above-mentioned project has been developed by the student(s) under my guidance and supervision.
2. The project addresses a real-world use case relevant to our organization.
3. The student(s) have demonstrated the ability to **successfully deploy the solution** in a functional or pilot-ready form.
4. The developed project has the potential to be adopted/implemented for the intended purpose within our organization.
5. All intellectual property rights, confidentiality, or proprietary rights, if applicable, are governed by our internal policies and this document does not transfer any such rights.

**Industry Mentor Details:**

* **Name:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* **Designation:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* **Organization:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* **Contact Number:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* **Email ID:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Signature & Seal:**

**INDEX**

|  |  |  |
| --- | --- | --- |
|  | Abstract | Page No. |
|  | Introduction (description of broad topic) |  |
|  | Motivation |  |
|  | Literature Review/Comparative work evaluation |  |
|  | Gap Analysis |  |
|  | Problem Statement |  |
|  | Objectives |  |
|  | Tools/platform Used |  |
|  | Methodology |  |
|  | Experimental Setup |  |
|  | Evaluation Metrics |  |
|  | Results And Discussion |  |
|  | Conclusion & Future Work |  |
|  | References |  |
|  | Annexure I: Plagiarism Report |  |
|  | Annexure II  Complete implementation code (optional) |  |

**ABSTRACT**

We are all aware of the importance of Closed-Circuit Television (CCTV) cameras. One of the most significant roles of a CCTV system is its capacity to transmit secure and timely information, which leads to increased security and situational awareness in security and operational contexts. Security cameras capture live and recorded footage that provides an actionable picture of an organization's security. Operators can use visual data to respond to an impending threat or operational problem, as well as plan ahead of time for a variety of occurrences. There is a constant rise of crimes due to various reasons like poverty, unemployment, religious conflicts, political issues, etc. There has been a significant rise in surveillance systems in India, as shown in the figure below. Even though recent decades have seen the emergence of CCTV surveillance as a mainstream crime prevention measure used around the world, one drawback of the regular CCTV cameras is that it only monitors a particular frame. Hence, in this project, we transformed the regular CCTV using artificial intelligence and machine learning algorithms, and implemented them in python, to add features like in-out movement detection-that tells whether a person is entering the area under surveillance or leaving it, and records this with time stamps, facial recognition of family members-it reads the input data and recognizes the faces added as input. It also detects noise in any frame, that is, captures any unnecessary movements in a frame, other than just monitoring the area for regular surveillance. Machine learning and AI is becoming more prevalent, as cameras are able to more accurately gather data and make predictions based on integrated analytical softwares that are being developed.

***KEYWORDS: CCTV, Surveillance, Machine Learning, Artificial Intelligence***

**Chapter 1**

**Introduction**

1. **Background of the project**

Urbanization, rapid economic liberalization, growing large-scale political turmoil, fierce conflicts, and inadequate and inappropriate policies are the basis of crime in urban areas. In addition, poverty and inequality caused by rising expectations and a sense of moral anger among some members of society contribute to increasing and increasing the level of crime. Structural adjustment programs pursued to promote economic growth, such as layoffs, shrinking civil servants, and selling civil servants, have led to increased poverty and inequality. One of the consequences of these programs is the sharp rise in unemployment, which is a major cause of crime epidemics and increases, especially in urban areas.

Not just this, Covid-19 pandemic and subsequent lockdown increased the crime rates in India itself by 28%. Unfortunately, the world we live in is becoming increasingly unsafe for all of us. We’ve seen a steady increase in the registered crime cases over the decade. Using [**Home Surveillance Camera**](https://www.vmukti.com/home-surveillance-camera/) we can detect crimes such as kidnapping, murder, burglary, etc. The emergence of closed-circuit television (CCTV) surveillance is a mainstream crime prevention measure used around the world. CCTV can dramatically increase the security of your property and keep your family protected. Once you get CCTV installed in your house, you no more need to worry about break-ins or burglary while you are away. Because whatever happens, it will still be recorded in your CCTV cameras. Thus, these cameras let you be, free of worry while you are away on a vacation or especially during business trips. It can very well help mitigate the crime in the first place and curb it. If a potential burglar sees your household is fully secured with CCTV in every nook and corner, it might as well put him off and because of the fear, he will deter from any crime that he earlier intended to do.

The importance of adequate security measures for homes and business spaces cannot be stressed enough. Many shops, commercial spaces, educational institutions, and public areas are now under the watchful eyes of surveillance systems. The recordings can be used to monitor and thwart crime. CCTV footages can also be produced as evidence in the court of law. The advancement in technology has also enabled sophisticated night vision cameras to capture every movement outside your house at night. This way you can be sure about the safety of you and your loved ones. With a sharp increase in setting up of CCTV cameras for commercial surveillance, the prices of CCTV systems have recorded a drastic decline. A smarter version of CCTV cameras will always add to the above benefits, as it can not only monitor the area under surveillance but also keep a check on the in-out movements, motion detection and family facial recognition, for a better safety provided to its customers, through a very simple and accessible GUI. Hence, the investment made on CCTV camera is a necessity.

Table 1. Existing systems

| **Factors** | **Evaluation Criteria** | **System A** | **System B** | **System C** |
| --- | --- | --- | --- | --- |
| Video Quality and Resolution | - Resolution (e.g., 1080p, 4K) | 4K | 1080p | 4K |
|  | - Low-light and night-vision capabilities | Excellent | Good | Very Good |
| Video Analytics | - Facial recognition | Yes | No | Yes |
|  | - Object detection | Yes | Yes | No |
| Real-time Monitoring & Alerts | - Real-time alerting to security personnel | Fast | Moderate | Fast |
|  | - Types of alerts (e.g., unauthorized access, intrusion) | Comprehensive | Limited | Comprehensive |
| Integration and Compatibility | - Integration with access control and other systems | Seamless | Limited | Moderate |
|  | - Compatibility with existing infrastructure | High | Moderate | Moderate |
| Scalability | - Ability to scale for additional cameras | Highly Scalable | Limited | Scalable |
| Remote Access & Management | - Remote access through mobile apps and web interfaces | Yes | Yes | Yes |
|  | - Remote management and configuration ease | Intuitive | Moderate | Intuitive |
| Storage & Data Management | - Storage options (cloud-based, on-premises) | Cloud & On-premises | Cloud | On-premises |
|  | - Data retention policies and search/retrieval capabilities | Flexible | Limited | Comprehensive |
| Privacy & Compliance | - Compliance with data protection regulations | Yes | Yes | Yes |
|  | - Features for masking sensitive areas and anonymizing individuals | Comprehensive | Limited | Comprehensive |
| Cost & ROI | - Upfront costs (installation, licensing) | Moderate | High | Moderate |
|  | - Potential ROI through improved security and efficiency | High | Moderate | High |
| Reliability & Maintenance | - Uptime and resilience to failures | Very Reliable | Reliable | Reliable |
|  | - Maintenance requirements (updates, calibration) | Low | Moderate | Low |
| User Interface & Ease of Use | - Intuitiveness for configuring and managing | Very Intuitive | Moderate | Intuitive |
| Customer Support & Training | - Quality of customer support and technical assistance | Excellent | Good | Moderate |

1. **MOTIVATION**

In recent decades, there have been a vast number of reasons contributing to the unexpectedly growing crimes. Urbanization, rapid economic liberalization, growing large-scale political turmoil, fierce conflicts, and inadequate and inappropriate policies can be listed as the basis of crime in urban areas. Moreover, crime rate has significantly increased due to current pandemic, and has made things worse for the security officials of all countries.

Even though we can never know the intentions of any person, but with the help of a CCTV camera, we can mitigate the risk of crime occurring around us, prevent the crime in the first place because if criminal would know that they are under surveillance, it might put them off and because of the fear, they will deter from any crime that they earlier intended to do.

In May 2021, Comparitech published a [report](https://www.comparitech.com/vpn-privacy/the-worlds-most-surveilled-cities/) on the use of CCTV cameras in 150 major cities across the globe. They found around 770 million cameras globally. The rising number of CCTV cameras in India is a cause of grave concern. Figure 1 shows that around 1.54 million cameras are spread among India's top 15 cities. New Delhi (5,51,500), Hyderabad (3,75,000), Chennai (2,80,000), and Indore (2,00,600) have the most surveillance cameras in the country. It is worth noting that almost 91.1% of CCTV cameras installed in the country are present only in these four cities.Also, CCTV Market Size is expected to Reach USD 46.52 Billion by 2030 at a 13.1% CAGR as suggested in a report by Market Research Future (MRFR).

The CCTVs eradicate the fear among the people in order to deter crime occurrence. Presence of the surveillance services assures the people that the surveyed areas deem more secure than the areas under no surveillance hence more people access the protected areas compared to the areas with no CCTV. The main goal of this project was to enhance the working of the currently used CCTV cameras, add in features that would increase the security of the households, are easy to access and are very effective in monitoring and mitigating any crimes.

The project adds in features like in-out detection, noise detection and facial recognition of family members along with monitoring the frame with time stamps.

**Chapter 2**

**LITERATURE REVIEW**

1. **Review of existing literature**

CCTV AS AN INVESTIGATIVE TOOL:

There has been extensive research on the value of CCTV for preventing crime, but little on its value as an investigative tool. The study by Matthew Ashby sought to establish how often CCTV provides useful evidence and how this is affected by circumstances, analysing 251,195 crimes recorded by British Transport Police that occurred on the British railway network between 2011 and 2015. CCTV was available to investigators in 45% of cases and judged to be useful in 29% (65% of cases in which it was available). Useful CCTV was associated with significantly increased chances of crimes being solved for all crime types except drugs/weapons possession and fraud. Images were more likely to be available for more-serious crimes, and less likely to be available for cases occurring at unknown times or in certain types of locations. Although this research was limited to offences on railways, it appears that CCTV is a powerful investigative tool for many types of crime.

INDOOR HOME SURVEILLANCE USING IOT:

The use of traditional CCTV to monitor the secured area have three limitations, which are requiring a huge volume of storage to store all the videos regardless there are intruders or not, does not notify the users immediately when there are motions detected, and users must always check the CCTV recorded videos regularly to identity any intruders. Therefore, a smart surveillance monitoring system is proposed to solve this problem by detecting intruders and capturing image of the intruder. Notifications will also be sent to the user immediately when motions are detected. This smart surveillance monitoring system only store the images of the intruders that triggered the motion sensor, making this system uses significantly less storage space.

ATM ROBBERY PREVENTION USING CCTV AND MACHINE LEARNING:

The idea if designing and implementation of the real-time ATM robbery prevention project came with the observation of real-time ATM robbery incidents around us. In this research paper by International Journal of Advanced Research, the project gives the alert at the instant of time when the thief is about to break the ATM machine.so, to overcome the drawbacks in the existing systems in our society. Whenever the thief is bringing the tools for the robbery into the ATM or when the thief is trying to use the tool to break, the CCTV camera on the ATM sense whether the person is bringing tools using the deep learning techniques and machine learning. Here the OpenCV and TensorFlow are used as the platform and the python language is used for the deep learning technics and Keras API were used for object detection.

IDENTIFYING MOVING BODIES WITH CCTV AND ML ALGORITHMS:

In this research paper publishes by IEEE, the project aims at observing that if there is a database of facial data present then the task of recognition boils down to comparison of each and every face detected from the video with every face saved in the database. Now this process involves capturing the faces beforehand. This is actually a very tedious job, so the database of images is created (/updated) as and when new faces come into the camera view. The labelling of the faces can be done at leisure (by a human) or not be done at all. The current system once deployed does not need a database of images to start with. It creates its own collection of images, and then tracks the future occurrences of those images.

REAL TIME DETECTION AND TRACKING OF PEDESTRIANS WITH CNN:

This research moves with the approach to match the extracted features of individual detections in subsequent frames, hence creating a correspondence of detections across multiple frames. The developed framework is able to address challenges like cluttered scenes, change in illumination, shadows and reflection, change in appearances and partial occlusions. However, total occlusion and similar persons in the same frame remain a challenge to be addressed. The framework is able to generate the detection and the tracking results at the rate of four frames per second, where CCTVs are spread over a huge range of area.

**Table 2. LITERATURE REVIEW/COMPARITIVE WORK**

| Project Title | Objectives | Technologies Used | Outcomes and Findings |
| --- | --- | --- | --- |
| Smart City Surveillance | Enhance urban safety and security | AI-powered video analytics | Reduced crime rates, faster incident response |
| System in City A | Monitor traffic flow and detect accidents | IoT sensors, facial recognition | Improved traffic management, enhanced safety |

|  | Prevent and investigate criminal activities | Cloud storage, license plate recognition | Increased arrest rates, streamlined investigations |
| --- | --- | --- | --- |
| AI-Driven Retail | Monitor customer behavior and store performance | Computer vision, behavior analytics | Optimized store layout, personalized shopping experiences |
| Surveillance in Mall B | Detect shoplifting and suspicious behavior | People counting, heatmap analysis | Reduced theft incidents, improved security |

|  | Analyze shopping patterns for marketing insights | Point-of-sale integration, customer tracking | Higher sales conversion, targeted marketing campaigns |
| --- | --- | --- | --- |
| Smart School Security | Enhance school campus safety | Access control systems, intrusion detection | Rapid response to threats, safer learning environment |
| System in School C | Monitor unauthorized entry and exit | RFID technology, real-time alerts | Decreased incidents of trespassing and bullying |

|  | Identify potential security breaches | Facial recognition, perimeter monitoring | Increased accountability, reduced external threats |
| --- | --- | --- | --- |
| IoT-Integrated Traffic | Improve traffic flow and congestion management | Vehicle detection sensors, smart signals | Reduced traffic congestion, shorter commute times |
| Management in City D | Optimize traffic signal timings | Real-time data analytics, cloud integration | Enhanced traffic efficiency, improved air quality |

1. **GAP ANALYSIS**

From the numerous researches done for enhancement of CCTV by applying machine learning algorithms, artificial intelligence, internet of things and convolutional neural network, the resultant CCTV camera is always a stronger product than the traditional version which just monitors a particular frame. But all these projects are carried out on external safety measures such as traffic management, security of borders, bank robbery and theft, but not many researches are carried out for safety at internal household level, or, they do not provide the best set of functionalities required for a safer household. Our project covers facial recognition of family members, in and out movement detection, motion detection in frame as well as monitoring the entire frame. Hence, we have covered a good subset of requirements, all available under one place, which will help the consumers feel more secure while they are away, and the easily accessible recordings with the help of a simple GUI will add up to the pros of the project.

1. **PROBLEM STATEMENT**

The closed-circuit television (CCTV) is one of the devices used to monitor the secured area for any intruders. The presence of surveillance cameras can act as a strong deterrent to criminals and thieves, as it is possible to identify people and track their movements, using the recorded footages.

The use of traditional CCTV to monitor the secured area, but they have their own set of limitations, some of which are requiring a huge volume of storage to store all the videos regardless there are intruders or not, does not notify the users when there are motions detected, and users must always check the CCTV recorded videos regularly to identity any intruders or unusual event occurring in the area and manually check for each mishappening. This creates a need of smarter CCTV cameras, which can ease the user’s task by automating most of these tasks. Such a camera would prove to be better in terms of providing security, mitigating risks of crime, preventing crimes and also monitoring and recording the footage if the crime occurs.

1. **OBJECTIVES**

The features include:

1. **Monitor** – monitors the area under surveillance.
2. **Identify** – Identifies the family members.
3. **Noise** **Detection**– Finds any motion in the frame.
4. **In** **Out** **Detection** – Finds who enters and exits.

The objective is to overcome the gap from various projects and to create a smart CCTV camera for households that can add a sense of security and help mitigate the crimes. The recorded footages can be considered as evidence, provided a crime occurs. The use of machine learning algorithms is done to make it smarter, and a GUI is created using tkinter in python to make it accessible and easy to understand for its users.

**CHAPTER 3: METHODOLOGY (NO PAGE LIMIT)**

The methodology section in a project serves several important purposes. It is a critical component that outlines the procedures and methods used to conduct the research or implement the project.

3.1 **Overall architecture /Flow chart** : describing the various modules in the project & interactions between various components. It must be diagram based.

The overall architecture of a project refers to the high-level design and structure that outlines how different components and modules of the project interact with each other. The specifics of the architecture will depend on the nature of the project, whether it's a software application, a machine learning system, a website, or another type of project. tic.

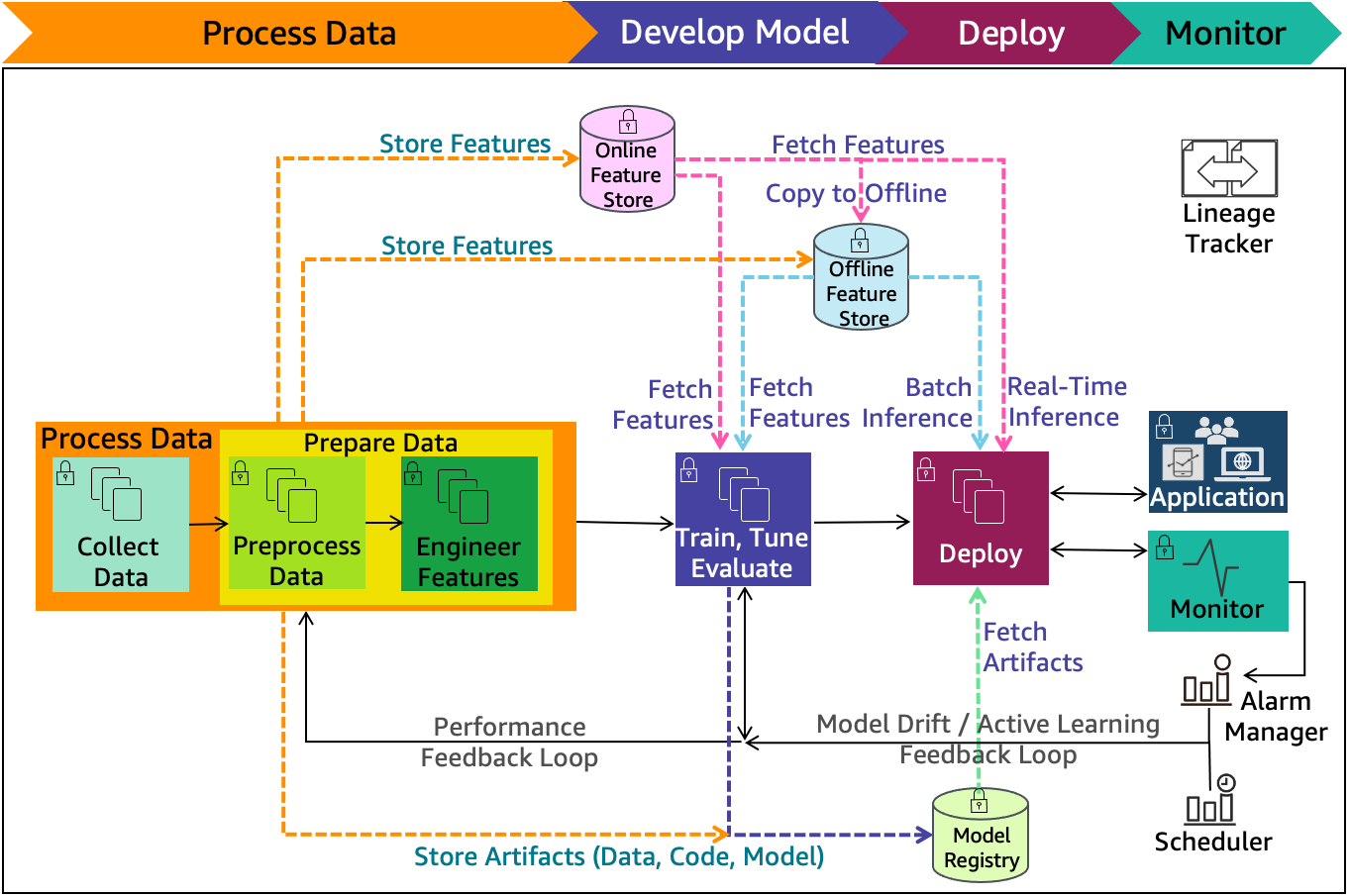


Figure 1. Figure Description

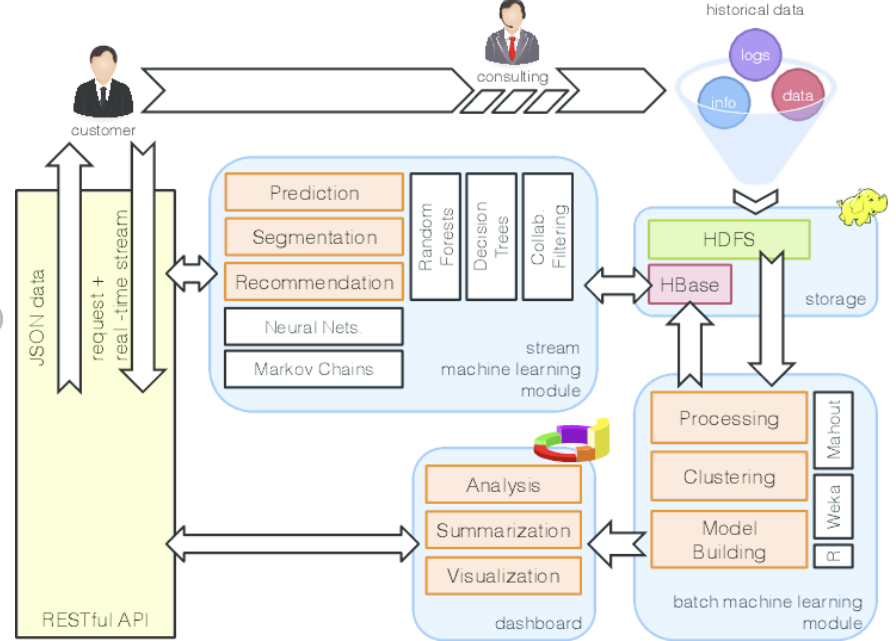


Figure 2. Describe the diagram in details

**3.2 Data Description**

**Data Source**: Describe the source of your data. It could be a specific dataset, a collection of documents, survey responses, experimental results, or any other relevant source.

Include details about the origin, organization, or platform from which the data was obtained.

**Data Collection Process**: Explain how the data was collected. Was it through surveys, experiments, observations, web scraping, sensor readings, or another method?

Provide a step-by-step explanation of the data collection process, including any tools or instruments used.

**Data Type:** Specify the type of data you are working with. Is it numerical, categorical, textual, time-series, or a combination of different types?

Highlight the key features that characterize your data.

Data Size: Mention the size of your dataset. Include details such as the number of records, observations, or instances.

If applicable, provide information on the dimensionality of the data (e.g., number of variables).

**Data Format:** Describe the format of the data files. For example, is the data stored in CSV, Excel, JSON, XML, database tables, or other formats?

Include information on how the data is structured and organized.

**Data Preprocessing:** Explain any preprocessing steps applied to the raw data. This may include data cleaning, handling missing values, scaling, normalization, or any transformations.

If you performed feature engineering, briefly describe the techniques used.

**Data Sampling (if applicable):** If you used a subset of the data for specific analyses, experiments, or model training, explain the sampling process.

Provide the rationale for the chosen sampling strategy.

**Data Quality Assurance**: Discuss how you ensured the quality and integrity of the data. This could involve data validation checks, outlier detection, or any other quality control measures.

If there are known limitations or challenges with the data quality, acknowledge them.

**Data Variables:** List and define the variables present in the dataset. Include both independent and dependent variables, as well as any control variables.

Specify the units of measurement and any relevant scales.

**Data Distribution and Summary Statistics**: Provide an overview of the distribution of key variables. This may include summary statistics such as mean, median, standard deviation, and quartiles.

Consider including visualizations like histograms or box plots for a better understanding of the data distribution.

**3.3 Exploratory data Analysis (if applicable)**

Exploratory Data Analysis (EDA) is a critical phase in the data analysis process that involves summarizing and visualizing key characteristics of a dataset to gain insights and identify patterns. Here are key components and techniques that should be part of the Exploratory Data Analysis:

**Summary Statistics:** Calculate and present basic statistical measures such as mean, median, mode, range, and standard deviation for each variable.

Use tools like Python's Pandas or R to generate summary statistics.

**Data Distribution**: Visualize the distribution of each variable using histograms, box plots, or kernel density plots.

Identify outliers and assess the skewness or kurtosis of the data.

**Correlation Analysis:** Compute correlation coefficients between pairs of variables to understand the strength and direction of relationships.

Create a correlation matrix and visualize it using a heatmap.

**Pairwise Scatter Plots:** Generate scatter plots for pairs of variables to explore relationships and identify potential patterns or trends.

Use different colors or shapes to highlight different categories or classes.

**Categorical Variable Exploration**: For categorical variables, create bar charts or count plots to visualize the distribution of categories.

Explore the relationships between categorical and numerical variables using box plots or violin plots.

**Missing Values Analysis**: Identify and quantify missing values in the dataset.

Visualize the patterns of missing data using heatmaps or other appropriate visualizations.

**Feature Engineering:** Explore opportunities for feature engineering based on domain knowledge or patterns observed during EDA.

Create new features or transform existing ones to better suit the analysis.

**Data Transformation:** Assess the need for data transformation, such as log transformations or scaling, to meet the assumptions of statistical tests or to improve model performance.

**Outlier Detection:** Use box plots, scatter plots, or statistical methods to identify potential outliers.

Decide whether to remove outliers based on the context of the analysis.

**Time Series Analysis (if applicable):** If the data involves a temporal component, analyze time trends using line plots, seasonal decomposition, autocorrelation plots, or other time series techniques.

Dimensionality Reduction:

Apply dimensionality reduction techniques, such as Principal Component Analysis (PCA) or t-SNE, to visualize high-dimensional data in lower dimensions.

**Interactive Visualizations:** Use interactive visualizations (e.g., Plotly, Bokeh) to allow users to explore the data dynamically.

Create dashboards or interactive widgets for a more engaging exploration experience.

**Data Slicing and Dicing:** Segment the data based on different criteria (e.g., time periods, geographical regions) to uncover patterns within specific subsets.

**Data Profiling:** Conduct data profiling to understand the data types, unique values, and basic characteristics of each variable.

Identify potential data quality issues or anomalies.

**Data Presentation:** Clearly present your findings through well-annotated visualizations, tables, and narratives.

Highlight interesting patterns or trends discovered during the exploration.

**Hypothesis Testing (if applicable):** If applicable, perform hypothesis tests to investigate relationships or differences between groups in the data.

**3.4** **Procedure /Development Life Cycle (depends on type of project)**

*Describe how various steps of development life cycle in context of your project were executed.*

**For Machine learning projects:**

* Data Collection: Gather text data from social media platforms, ensuring a balanced distribution of sentiments.
* Data Preprocessing: Clean and preprocess the text data, including tokenization, stemming, and removal of stop words.
* Feature Extraction: Utilize NLP techniques to convert text into numerical features, such as TF-IDF vectors or word embeddings.
* Model Training: Employ supervised learning algorithms, such as a support vector machine (SVM) or a neural network, to train the sentiment analysis model.
* Model Evaluation: Evaluate the model using a separate test dataset, employing metrics like accuracy, precision, recall, and F1 score.
* Fine-Tuning: Iteratively fine-tune the model based on evaluation results to enhance performance.
* Deployment: If deemed satisfactory, deploy the model for real-time sentiment analysis.

1. **Details of tools, software, and equipment utilized.**

**PLATFORM USED**

For this project, we have used various latest technologies which will be evaluated in this chapter with every detail of why it is used.

PROGRAMMING LANGUAGE: **PYTHON**

We have used Python language as it is relatively new as compared to other languages like Java, C++, etc and comes with so many features. We can perform Machine Learning, Computer Vision, Artificial Intelligence, etc with python and construction of GUI application is also easily achieved in Python.

[Python](https://www.geeksforgeeks.org/python-programming-language/) is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. There are two major Python versions: Python 2 and Python 3

Reasons for Selecting this language:

1. Short and Concise Language.
2. Easy to Learn and use.
3. Good Technical support over Internet
4. Many Packages for different tasks.
5. Run on Any Platform.
6. Modern and OOP language

Some specific features of Python are as follows:

1. An interpreted (as opposed to compiled) language. Contrary to e.g. C or Fortran, one does not compile Python code before executing it. In addition, Python can be used interactively: many Python interpreters are available, from which commands and scripts can be executed.
2. A free software released under an open-source license: Python can be used and distributed free of charge, even for building commercial software.
3. Multi-platform: Python is available for all major operating systems, Windows, Linux/Unix, MacOS X, most likely your mobile phone OS, etc.
4. A very readable language with clear non-verbose syntax.
5. A language for which a large variety of high-quality packages are available for various applications, from web frameworks to scientific computing.
6. A language very easy to interface with other languages, in particular C and C++.
7. Some other features of the language are illustrated just below. For example, Python is an object-oriented language, with dynamic typing (the same variable can contain objects of different types during the course of a program).

The features included in the project are:

1. **Monitor** – monitors the area under surveillance.
2. **Identify** – Identifies the family members.
3. **Noise** **Detection**– Finds any motion in the frame.
4. **In** **Out** **Detection** – Finds who enters and exits.

Each feature is discussed in detail in the next section, the methodology used in the features is elaborated and the process model used for the completion of the project is also mentioned below.

1. **ENVIRONMENTAL SETUP**

SOFTWARE REQUIREMENTS

Below are the requirements to run this software :

1. Windows/Linux/Mac OS any version, hence it can run on any platform.
2. Python3, it needs python to be installed in system to run successfully.
3. Packages in python -
   1. openCV
   2. skimage
   3. numpy
   4. tkinter

**HARDWARE REQUIREMENTS**

In terms of hardware requirements there is not much required at all but still below requirements are must:

1. Working PC or Laptop
2. Webcam with drivers installed
3. Flashlight/ LED if using this at night.

**PLATFORMS ALREADY TESTED ON:**

It is tested on Linux Mint, Linux Ubuntu, Windows 7 and Windows 10.

**Chapter 4**

**Implementation**

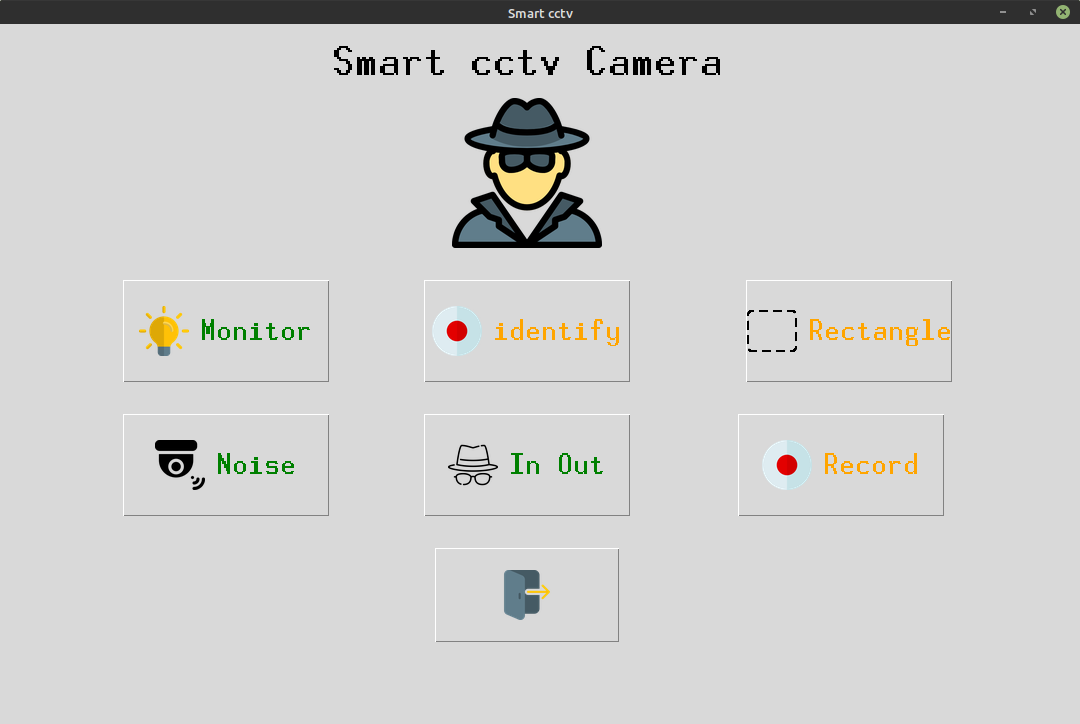
1. **Detailed explanation of how the project was implemented.**
2. **Description of algorithms, code snippets, or design diagrams.**
3. **Discussion of any challenges faced during implementation and their solutions.**

**Students will provide content as per their project work as per the pointers**

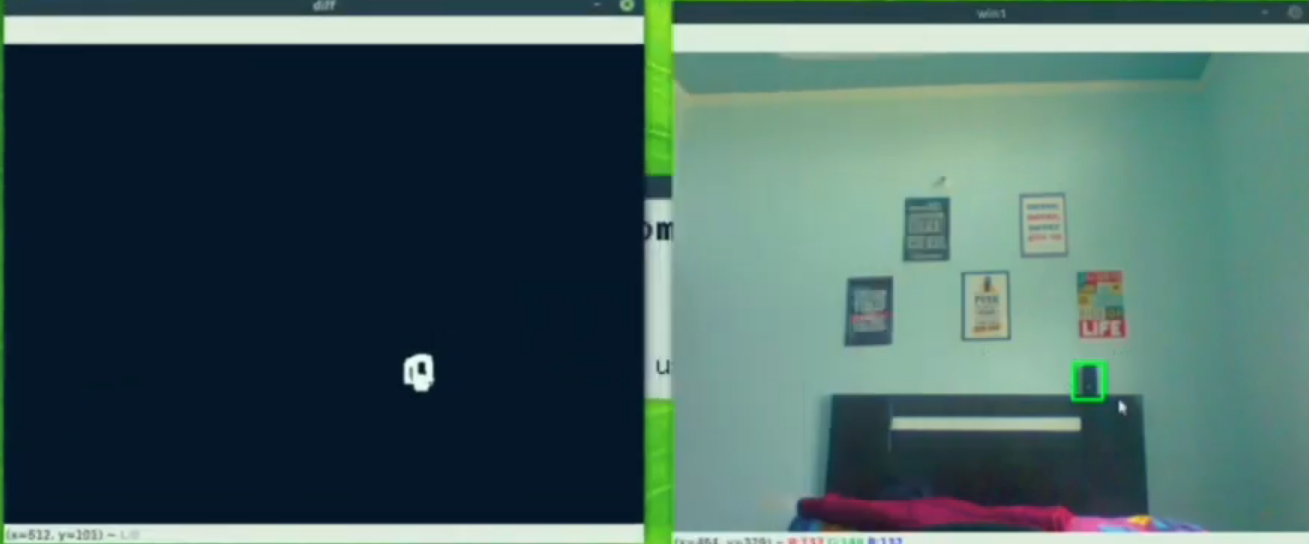
**Chapter 5**

**RESULTS AND DISCUSSIONS**

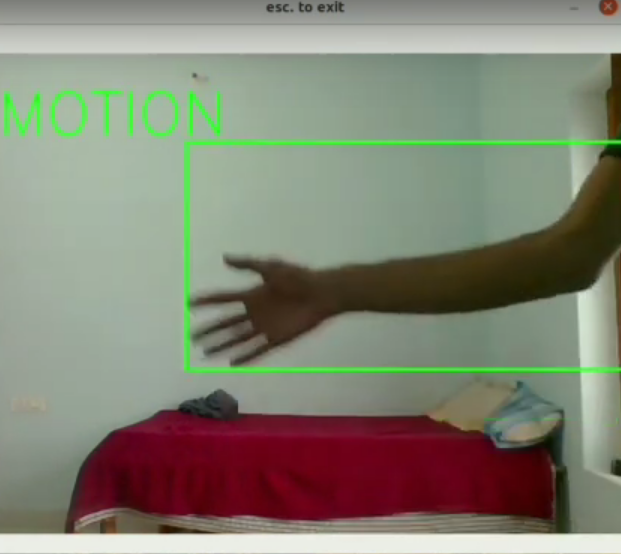
**THE GUI:**



**MONITOR FEATURE:**

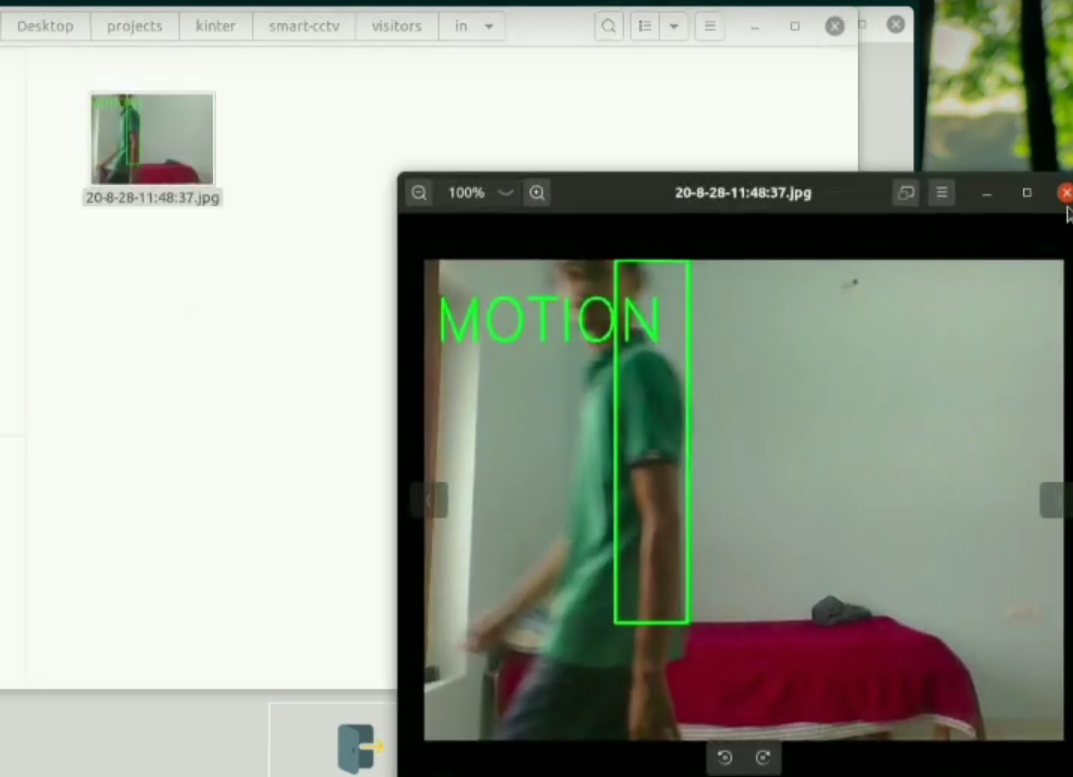
Correctly detecting the missing speaker from the frame.

**2.NOISE DETECTION FEATURE:**

Detecting hand motion correctly.



**3.IN-OUT MOVEMENT DETECTION:**

Saves the image locally as – entered the room.

**Chapter 6**

**FUTURE WORK**

Theproject is working well for all the features including- monitoring, facial detection, noise recognition and in-out movement detection. The future work can include enhancing the model by using a better face recognition approach like Dlib, as haarcascade classifier is not very accurate at all times. The project is scaled for a very limited user base, so it can only detect a little motion and only a couple entries and exits simultaneously. It can be scaled up for more number of users and thus can be used in other domains like bank security management, school security, etc. Better technology for night vision can be implemented as it is not appropriate for darker regions.

**CONCLUSION**

In recent decades, there have been a vast number of reasons contributing to the unexpectedly growing crimes. Urbanization, rapid economic liberalization, growing large-scale political turmoil, fierce conflicts, and inadequate and inappropriate policies can be listed as the basis of crime in urban areas. Moreover, crime rate has significantly increased due to current pandemic, and has made things worse for the security officials of all countries.

The closed-circuit television (CCTV) is one of the devices used to monitor the secured area for any intruders. But the traditional CCTVs have their own set of flaws, which make them less effective for securing the area. This project tries to implement Machine Learning algorithms to enhance the working of traditional CCTVs, by adding functionalities:

1. **Monitor** – monitors the area under surveillance.
2. **Identify** – Identifies the family members.
3. **Noise** **Detection**– Finds any motion in the frame.
4. **In** **Out** **Detection** – Finds who enters and exits.

This will help make a stronger system for security concerns, and will make the users feel more secure when they are not at home. It will not only mitigate the risks of crime occurrence, but also capture anything and everything that can be considered as a proof against the criminal , provided the crime takes place.

**REFERENCES**

1. Kim, J., & Kim, H. J. (2019). A Smart CCTV System Based on Deep Learning for Real-time Human Detection in a Construction Site. Automation in Construction, 106, 102873.
2. Gupta, A., Kumar, S., & Chakraborty, S. (2020). IoT Based Smart Surveillance System Using CCTV Camera and Raspberry Pi. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.
3. Zhou, S., & Huang, R. (2017). Smart Video Surveillance System Based on IoT and Cloud Computing. Future Generation Computer Systems, 76, 319-326.
4. Sankaranarayanan, S., Al-Maadeed, S., & Trivedi, M. M. (2019). Smart CCTV Video Analysis and Anomaly Detection in Traffic and Transportation: A Survey. IEEE Transactions on Intelligent Transportation Systems, 21(1), 256-269.

**ANNEXURE I:**

**Plagiarism Declaration Certificate**

Title of Work:

Submitted By:

Institution: K. R. Mangalam University, Gurugram

Department:

Date of Submission:

I hereby declare that the work entitled ”[Insert Title of Work]” submitted for academic evaluation and research purposes is my original work. I confirm that:

* I have acknowledged and properly cited all sources, references, and data included in this work.
* This work does not contain any material previously published, written, or prepared by another person, except where due acknowledgment has been made.
* I understand that plagiarism is an academic offense and a violation of research ethics. Any breach of this declaration may lead to disciplinary action as per university policies.
* I have used appropriate referencing techniques and maintained academic integrity throughout this work.

I affirm that the submitted work has normal plagiarism less than 10% and free from AI content.

⸻

Signature of Student: \_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_

**COMPLETE IMPLEMENTATION**

The coding part in below with specific modules discussed above.

Main.py

import tkinter as tk

import tkinter.font as font

from in\_out import in\_out

from motion import noise

from rect\_noise import rect\_noise

from record import record

from PIL import Image, ImageTk

from find\_motion import find\_motion

from identify import maincall

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.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-cutted-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)

btn7\_image = Image.open('icons/recording.png')

btn7\_image = btn7\_image.resize((50,50), Image.ANTIALIAS)

btn7\_image = ImageTk.PhotoImage(btn7\_image)

# --------------- Button -------------------#

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()

Monitors is divided into two modules.

1. find\_noise.py
2. spot\_diff.py

import cv2

from spot\_diff import spot\_diff

import time

import numpy as np

def find\_motion():

motion\_detected = False

is\_start\_done = False

cap = cv2.VideoCapture(0)

check = []

print("waiting for 2 seconds")

time.sleep(2)

frame1 = cap.read()

\_, frm1 = cap.read()

frm1 = cv2.cvtColor(frm1, cv2.COLOR\_BGR2GRAY)

while True:

\_, frm2 = cap.read()

frm2 = cv2.cvtColor(frm2, cv2.COLOR\_BGR2GRAY)

diff = cv2.absdiff(frm1, frm2)

\_, thresh = cv2.threshold(diff, 30, 255, cv2.THRESH\_BINARY)

contors = cv2.findContours(thresh, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)[0]

#look at it

contors = [c for c in contors if cv2.contourArea(c) > 25]

if len(contors) > 5:

cv2.putText(thresh, "motion detected", (50,50), cv2.FONT\_HERSHEY\_SIMPLEX, 2, 255)

motion\_detected = True

is\_start\_done = False

elif motion\_detected and len(contors) < 3:

if (is\_start\_done) == False:

start = time.time()

is\_start\_done = True

end = time.time()

end = time.time()

print(end-start)

if (end - start) > 4:

frame2 = cap.read()

cap.release()

cv2.destroyAllWindows()

x = spot\_diff(frame1, frame2)

if x == 0:

print("runnig again")

return

else:

print("found motion sending mail")

return

else:

cv2.putText(thresh, "no motion detected", (50,50), cv2.FONT\_HERSHEY\_SIMPLEX, 2, 255)

cv2.imshow("winname", thresh)

\_, frm1 = cap.read()

frm1 = cv2.cvtColor(frm1, cv2.COLOR\_BGR2GRAY)

if cv2.waitKey(1) == 27:

break

return

spot\_diff.py

import cv2

import time

from skimage.metrics import structural\_similarity

from datetime import datetime

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

identify.py

import cv2

import os

import numpy as np

import tkinter as tk

import tkinter.font as font

def collect\_data():

name = input("Enter name of person : ")

count = 1

ids = input("Enter ID: ")

cap = cv2.VideoCapture(0)

filename = "haarcascade\_frontalface\_default.xml"

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]

cv2.imwrite(f"persons/{name}-{count}-{ids}.jpg", roi)

count = count + 1

cv2.putText(frm, f"{count}", (20,20), cv2.FONT\_HERSHEY\_PLAIN, 2, (0,255,0), 3)

cv2.imshow("new", roi)

cv2.imshow("identify", frm)

if cv2.waitKey(1) == 27 or count > 200:

cv2.destroyAllWindows()

cap.release()

train()

break

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

def maincall():

root = tk.Tk()

root.geometry("480x100")

root.title("identify")

label = tk.Label(root, text="Select below buttons ")

label.grid(row=0, columnspan=2)

label\_font = font.Font(size=35, weight='bold',family='Helvetica')

label['font'] = label\_font

btn\_font = font.Font(size=25)

button1 = tk.Button(root, text="Add Member ", command=collect\_data, height=2, width=20)

button1.grid(row=1, column=0, pady=(10,10), padx=(5,5))

button1['font'] = btn\_font

button2 = tk.Button(root, text="Start with known ", command=identify, height=2, width=20)

button2.grid(row=1, column=1,pady=(10,10), padx=(5,5))

button2['font'] = btn\_font

root.mainloop()

return

in\_out.py

import cv2

from datetime import datetime

def 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

motion.py

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

Finding noise in the rectangle.

import cv2

donel = 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, doner

cap = 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

At-last this is most required feature which is recording:

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