

ADVANCED VIDEO SURVEILLIANCE SYSTEM BY USING COMPUTER VISION

A Project Report Phase – II

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Electronics and Communication Engineering

By

CH. SRINADH (39130090)

CH. PRAVEEN KUMAR (39130091)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICAITON ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Final Year Project Report is the bonafide work of CH. SRINADH (REG NO: 39130090) AND CH. PRAVEEN KUMAR (REG NO: 39130091) who have done the project entitled "ADVANCE VIDEO SURVEILLANCE SYSTEM BY USING COMUPTER VISION" under our supervision from Nov 2023 to April 2023.

Internal Guide

Dr. E. ANNADEVI M.E., Ph.D.

Head of the Department

Dr. T. RAVI, M.E., Ph.D.,

Submitted for Viva voce Examination held on

24 - 04 - 2023

Internal Examiner

External Examiner

DECLARATION

We, **CH. SRINADH** (REG NO: 39130090), **CH. PRAVEEN KUMAR** (REG NO: 39130091), hereby declare that the Project Report on “**ADVANCED VIDEO SURVEILLIANCE SYSTEM BY USING COMPUTER VISION**” done by us under the guidance of **Dr.E. ANNADEVI. M.E. Ph.D.**, at Sathyabama Institute of Science and technology is submitted in partial fulfilment of the requirements of the award of Bachelor of Engineering degree in **Electronics and Communication Engineering**.

DATE: 24 – 04 – 2023

PLACE : CHENNAI

Signatures of the candidates

1. 

2. 

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **the Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D., Professor & Dean, School of Electrical and Electronics** and **Dr. T. RAVI, Ph.D., Professor & Head, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our **Project Guide Dr. E. ANNADEVI. M.E. Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved the way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Safety and security are major concerns in the modern era. People employ security mechanisms to safeguard their property be it home or a company. Present security systems involve the use of various sensors for motion detection and cameras for video surveillance i.e., Perimeter Intrusion Detection Systems. This project aims at providing one such idea to ensure safety and security of one's own property. In this paper we propose to provide a Smart CCTV Surveillance system with intrusion detection. Multiple USB cameras are installed at various locations for live streaming and monitoring purposes. This system performs face recognition as an authentication procedure and alerts the owner when an unknown face is detected by sending an email with the snapshot of unknown face and an SMS. Live feed from multiple cameras can be viewed on smartphones and computers.

TABLE OF CONTENTS

CHAPTER.NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	vii
1.	INTRODUCTION	1
	1.1 INTROCUCTION	3
	1.2 OBJECTIVES	5
	1.3 DESCRIPTION	2
2.	LITERATURE SURVEY	6
3.	SYSTEM ANALYSIS	10
	3.1 EXISTING SYSTEM	10
	3.2 PROPOSED SYSTEM	10
	3.3 REQUIREMENT SPECIFICATION	11
4.	SYSTEM DESIGN	16
	4.1SYSTEM ARCHITECTURE	16
	4.2 USE-CASE DIAGRAM	17
	4.3 ACTIVITY DIAGRAM	18
	4.4 DATA FLOW DIAGRAM	19
5.	MODULE DESCRIPTION	21
	5.1 MONITOR	21
	5.2 IDENTIFY	21
	5.3 USING LBPH FOR FACE RECOGNITION	28
6.	TESTING	29
	6.1 SYSTEM TESTING	29
	6.2 TESTING TECHNIQUES	30
	6.3 SOFTWARE TESTING STRATEGIES	33
7.	RESULTS AND DISCUSSIONS	42
8.	CONCLUSION	
9.	REFERENCES	50

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.1	SYSTEM ARCHITECTURE	16
4.2	USE-CASE DIAGRAM	17
4.3	ACTIVITY DIAGRAM	18
4.4	DATA FLOW DIAGRAM	19
4.5	COLLABORATION DIAGRAM	19
4.6	CLASS DIAGRAM	20
7.1	IDENTIFYING NAME OF THE PERSON	35
7.2	MAIN GUI WINDOW	36
7.3	UNKNOWN IDENTIFICATION	37
7.4	NO MOTION DETECTION	38
7.5	MOTION DETECTION	40
7.6	RECTANGLE - NO MOTION DETECTION	42
7.7	RECTANGLE - MOTION DETECTION	44
7.8	SPOT THE DIFFERECNCE DETECTION	46
7.9	RECORDING USING DATE AND TIME	48

Chapter - 1

INTRODUCTION

1.1 Introduction

In an age which bears witness to a proliferation of Closed-Circuit Television (CCTV) cameras for security and surveillance monitoring, the use of image processing and computer vision techniques which were provided as top end bespoke solutions can now be realized using desktop PC processing. Commercial Video Motion Detection (VMD) and Intelligent Scene Monitoring (ISM) systems are becoming increasingly sophisticated, aided, in no small way, by a technology transfer from previously exclusively military research sectors. Image processing is traditionally concerned with pre-processing operations such as Fourier filtering, edge detection and morphological operations.

Computer vision extends the image processing paradigm to include understanding of scene content, tracking and object classification. Examples of computer vision applications include Automatic Number Plate Recognition (ANPR), people and vehicle tracking, crowd analysis and model based vision. Often image processing and computer vision techniques are developed with highly specific applications in mind and the goal of a more global understanding computer vision system remains, at least for now, outside the bounds of present technology. With the increasing availability of high-performance computing systems, the field of computer vision is advancing rapidly. Deep learning techniques have revolutionized computer vision, allowing for more complex tasks such as object recognition and segmentation, image captioning, and even autonomous driving. These advances have been made possible by the availability of large-scale datasets, such as ImageNet and COCO, and the development of powerful neural network architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).

The use of computer vision techniques is not limited to security and surveillance applications. In healthcare, computer vision is being used for medical imaging analysis, disease diagnosis, and surgical assistance. In retail, computer

vision is used for inventory management, customer tracking, and behavior analysis. In agriculture, computer vision is being used for crop monitoring and yield prediction. The potential applications of computer vision are vast, and the technology.

As computer vision technology becomes more advanced, there are concerns about its impact on privacy and security. The widespread use of surveillance cameras and facial recognition technology has raised questions about the ethical implications of these technologies, as well as the potential for misuse. There is a need for greater transparency and accountability in the development and deployment of computer vision systems to ensure that they are used in a responsible and ethical manner. Another challenge in computer vision is the interpretation and explain ability of results. Deep learning models are often described as "black boxes" because it is difficult to understand how they arrive at their decisions. This lack of interpretability is a significant barrier to the adoption of computer vision in critical applications such as healthcare and autonomous vehicles. There is a need for research on explainable AI, which aims to develop techniques for interpreting and explaining the decisions made by deep learning models.

In addition to technical challenges, there are also societal challenges associated with the use of computer vision technology. One of the main concerns is the potential for bias in computer vision systems. Biases can arise from the data used to train the models, as well as from the design of the models themselves. There is a need for greater diversity and representation in the datasets used to train computer vision systems, as well as for techniques to mitigate bias in the design of the models. The future of computer vision is likely to be shaped by advances in related fields such as robotics, natural language processing, and augmented reality. As these technologies converge, we can expect to see new applications of computer vision in areas such as human-robot interaction, immersive gaming, and virtual and augmented reality. Intelligent surveillance systems can provide valuable insights into the behavior and activities of the individuals being monitored. For example, behavior analysis algorithms can be used to identify suspicious or abnormal behavior, such as loitering or looking

around nervously.

1.2 Objectives

The main purpose of this system is to improve the awareness of security personnel and decision makers by collecting real-time information automatically. The objective of this project is to develop a system that monitors the area in which it is being implemented. An Intelligent surveillance system is applicable in the area where no one is permissible to enter, also where we need to detect if any motion has been done. The system raises an alarm whenever unacceptable movements are detected. Hence, the system has the ability to detect mobile objects in the scene and to classify their movements (as allowed or disallowed).

The development of an intelligent surveillance system involves several technical components and considerations. Some of the key areas of focus include.

Sensor Technology: The system requires sensors to detect and capture data about the environment. Common sensors used in surveillance systems include video cameras, motion sensors, and acoustic sensors. These sensors need to be strategically placed in the area being monitored to ensure optimal coverage.

Data Analytics: Once the data is collected, the system needs to be able to analyze it in real-time. This requires sophisticated data analytics tools such as machine learning algorithms, pattern recognition, and anomaly detection.

Communication and Networking: The surveillance system needs to be able to communicate with other devices and systems. This includes sending alerts and notifications to security personnel or other relevant parties.

Security and Privacy: The system must be designed with robust security measures to ensure that sensitive data is not compromised. This includes encryption, access controls, and secure data storage. The system should also be designed to protect the privacy of individuals in the area being monitored.

User Interface: The system must have an intuitive and user-friendly interface for security personnel and decision-makers to monitor and manage the system. This

includes the ability to configure settings, view alerts and notifications, and access

Overall, the development of an intelligent surveillance system requires a multi-disciplinary approach that involves expertise in areas such as data analytics, sensor technology, networking, security, and user interface design. Intelligent surveillance systems can be used in a wide range of applications, including public safety, transportation, and industrial automation. For example, in public safety, intelligent surveillance systems can be used to monitor public areas, such as airports or train stations, to detect suspicious behavior and potential security threats. In transportation, intelligent surveillance systems can be used to monitor traffic flow and detect accidents or incidents, allowing for faster response times and improved safety. In industrial automation, intelligent surveillance systems can be used to monitor production lines and detect defects or malfunctions, allowing for more efficient and effective manufacturing processes.

The use of computer vision algorithms in intelligent surveillance systems enables them to perform a range of advanced functions, such as object tracking, facial recognition, and behavior analysis. These functions can help to improve the accuracy and efficiency of the surveillance system, as well as provide valuable insights into the behavior and activities of the individuals being monitored. Computer vision algorithms are used in intelligent surveillance systems to analyze video data and perform a range of advanced functions. These functions include object tracking, which allows the system to track the movement of objects or individuals across the surveillance zone, and facial recognition, which can be used to identify individuals based on their facial features. Other functions include behavior analysis, which can help to identify suspicious or abnormal behavior, and activity recognition, which can be used to classify different types of activity, such as walking, running, or standing. The use of computer vision algorithms can help to improve the accuracy and efficiency of intelligent surveillance systems by automating many of the tasks that would otherwise need to be performed manually.

For example, object tracking algorithms can track the movement of objects or individuals across multiple cameras, allowing the system to build a more complete picture of the surveillance zone. Similarly, facial recognition algorithms can be used to identify individuals in real-time, providing valuable insights into who is

entering and leaving the monitored area.

1.3 Description

This is a python GUI application that can run on any operating system, uses a web cam, and has a number of features that are not in normal CCTV, discussed in detail below pages. This is a Project built using the latest Programming Language and highly evolving Computer Science field which is “ComputerVision”. This means this project allow computer to watch or in other words it gives vision capability to computers. Computer Vision is an interdisciplinary field of study that enables machines to interpret and understand digital images and videos from the world around us. By using techniques such as image processing, pattern recognition, and machine learning, Computer Vision allows machines to acquire, analyze, and understand visual information in a way that is similar to human vision.

Python is a high-level programming language that is widely used in Computer Vision applications due to its simplicity, ease of use, and availability of libraries such as OpenCV (Open Source Computer Vision Library). OpenCV is a popular library for Computer Vision applications, providing a wide range of tools and functions for image and video processing, feature detection, and object recognition. The use of a webcam in this GUI application allows for real-time video feed and monitoring, enabling the system to capture and analyze video streams as they happen. This can be particularly useful in security applications, where real-time monitoring and alerts are crucial for detecting potential security risks or breaches.

The additional features that are not typically found in traditional CCTV systems can provide a range of benefits and advantages, such as object detection and tracking, no motion detection, and spot the difference detection. These features can help improve the accuracy and efficiency of the system, as well as reduce the likelihood of false alarms or missed events.

CHAPTER - 2

LITERATURE REVIEWS

Image Processing Based System for Intrusion Detection and Home Security Enhancement Sivakumar, Swetha, and R. Gomathi Bhavani This paper combines the above two methodologies and proposes a simple, cost effective, two-layered system. The system is proposed for the main doorway of the property and has been designed and demonstrated already. In the above system, an Arduino controller communicates with sensors and image processing tools of MATLAB to authorize entry using GSM module. This makes sure that only personnel recognized by owner database are given permission, limiting intrusions to a large probability.

Face Detection and tracking using Image processing on Raspberry Pi Bhanse, Vivek Kishor, and M. D. Jaybhaye The paper shows the examination how the assurance of the camera impacts the edge each seconds and respectably the perfect open door for stand up to acknowledgment. The image processing using OpenCV for face detection, face tracking and its recognition for automotive application is done here. We have used processor controller and common usb camera for controlling the electrical motor which in this way controls the motorized auto application. the viola- jones estimation for stand up to acknowledgment using adaboost library in this way structure relies upon adaboost computation and changed works faces haar-like highlights in the item part we have used python vernacular with opencv reinforce for the face distinguishing proof using haar course records gave by intel open source. we have shown the results on layout consistently impact on stand up to distinguishing proof Bhante.

CCTV footage, without the use of sensors. This system concentrates on object detection. The security personnel can be notified about the suspicious individual committing burglary using Real-time analysis of the movement of any human from CCTV footage and thus gives a chance to avert the same.

Smart IOT Surveillance Multi-Camera Monitoring System Husniza Razalli; Mohammed Hazim Alkawaz; Aizat Syazwan Suhemi The surveillance digital monitoring process market has changed aggressively over the last 15 years. This system only implements open source image processing methods to produce a monitoring system with customizable modules for video analytics (video content analysis) with the input of live video sources. This is to allow a wider range of camera models that can be used for this system. A real-time analysis module will assist in the use of the system to ease the user. The combined video stream and the module will help the user immensely for surveillance propose. Based on the result, the proposed system achieved a higher affordability level up to 95% with 90% usability compared to existing products.

Surveillance Camera using IoT and Raspberry Bandi Narasimha Rao; Reddy Sudheer Security remains as one of the most important aspects of smart cities. At present, due to the increased population and urbanization there is a huge demand for smart security system. Hence, the proposed system provides a secured data transmission. The live feeding is deployed by using Raspicam Remote App. The proposed model can be easily configured, where it allows us to test the image filters of the camera. Hence, the proposed surveillance camera using raspberry pi and IoT appears to be more superior to the conventional IP cameras. The combined video stream and the module will help the user immensely for surveillance propose. Based on the result, the proposed system achieved a higher affordability level up to 95% with 90% usability compared to existing products.

The smart surveillance system using cloud technology consists of several components such as cameras, sensors, storage devices, and cloud servers. The cameras and sensors are used to capture video footage and other data, which is then transmitted to the cloud servers for storage and analysis. The proposed system has a smaller size and high processing power. The proposed system has its advantages of being highly accurate, precise, bears low computation time and costs, and can be used for real-world applications.

IOT based smart surveillance monitoring by using model-based human action recognition Amir Reza Ebrahimi; Ahmad Reza Naghsh Nilchi; Amir Hassan Monadjemi; Mohammad Saeid Ehsani In recent years, with the rapid development in IP camera usage, massive video surveillance data are produced at an unprecedented speed. IP camera is one of the most important elements in Internet of things (IoT) and smart cities, in this era. IoT Based Smart Surveillance provides resources over the Internet and allows a plethora of applications to be deployed to provide services for different applications. Traditional solutions to deal with the big video data would require a large amount of computing and storage resources.

Gopal Deshmukh IoT based Smart Security and Surveillance System Gurusha Lulla; Abhinav Kumar; Govind Pole Internet of Things (IoT) is an escalating trend in today's world and has proved to be a game changer in the field of technology. Also, the system provides a warning to the person who has intentionally or unintentionally entered the property, so that they can step back in time, without triggering any alarm. With the help of a remote camera surveillance feature, the owner can monitor the surroundings of their property remotely. This system will issue an alert to the owner in case of a power failure as well. This system provides face recognition as an authentication procedure, in order to allow the valid entrants to enter into their property. All these features combined, create a flexible and reliable security system, which can be used in various properties, such as homes, offices, museums and so on.

A Smart IoT Security System for Smart-Home Using Motion Detection and Facial Recognition AKM Jahangir Majumder; Joshua Aaron Izaguirre The Internet of Things (IoT) combines the idea of technology providing value to our daily lives. One major category in which technology benefits us is the concept of security and privacy. Smartphones can act as a security alert system as the smartphone is the most widely used smart device. Recently, the use of Artificial Intelligence (AI)-integrated.

Development of Smart Surveillance System using Cloud for Security Abdul Wahab Usman Ullah Advancements in technology have increased the quality and performance in surveillance systems, yet the current security systems lack full coverage monitoring which is expensive as the customer needs to spend a great deal on configuration, installation and maintenance etc. The system is cost-effective in reducing human work, energy, time, and cognitive mistakes. Not only is this sort of system important for safety purposes in households, offices, etc., but also plays an incredibly significant role for the physically challenged people who cannot move properly.

Smart surveillance systems using cloud technology have revolutionized the security industry by providing cost-effective and efficient solutions for monitoring and surveillance. These systems use advanced technologies such as Artificial Intelligence (AI), Machine Learning (ML), and Internet of Things (IoT) to enhance the monitoring and surveillance capabilities of the traditional security systems. The smart surveillance system using cloud technology consists of several components such as cameras, sensors, storage devices, and cloud servers. The cameras and sensors are used to capture video footage and other data, which is then transmitted to the cloud servers for storage and analysis.

The cloud servers are equipped with AI and ML algorithms that analyze video footage and other data in real-time to detect any suspicious activities or anomalies. The system can also be configured to send alerts and notifications to the security personnel in case of any security breaches or unusual activities. One of the major advantages of the smart surveillance system using cloud technology is its cost-effectiveness. The system eliminates the need for expensive hardware and software installations, as all the data. To implement this system, you would need to set up your web camera and configure it to work with OpenCV. Then, you would use the "Video Capture" function to capture frames from the camera stream, and the "inwrite" function to write each frame to a file with a unique filename that includes the date and time.

CHAPTER – 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

There are many existing intruder detection systems for surveillance and security systems. They either need human intervention to detect intruder or need a long work for the installation and there is also a possibility of false alarms. In Surveillance system, the intruder is detected using the video recordable cameras which are already installed and stored in an external storage disk. But in this system needs a huge investment for installing, storing and monitoring the activities. Though the occurrence of activity is less, the footage is to be deleted, after examining by the owner. This leads to waste of man hour as he should watch the complete video, and this may lead to missing small details during analysis.

3.1 PROPOSED SYSTEM

The proposed system is a python GUI application that can run on any operating system, uses a webcam and has a number of features that are not in normal CCTV, Smart video surveillance is integrated with intelligent video movement detection analysis systems combined with the SMS notification system. Built in an impressive GUI, the proposed system supports the following features

- *Monitor* – allow detecting what thing is stolen from the frame
- *Identity* – Finds the family members(it has to be trained first)
- *Noise* – Finds the motions in the frame
- *In Out* – Finds who entered and Gone from the room.

The proposed system has a smaller size and high processing power. The proposed system has its advantages of being highly accurate, precise, bears low computation time and costs, and can be used for real-world applications.

3.2 REQUIREMENT SPECIFICATION

- **Hardware Requirements:**

Processor	: Pentium Dual Core 2.00GHZ
Hard disk	: 120 GB
RAM	: 2GB (minimum)
Keyboard	: 110 keys enhanced

- **Software Requirements:**

Operating system	: Windows7 (with service pack 1), 8, 8.1 and 10IDE
Software	: Anaconda
Language	: Python

3.3 PLATFORM SPECIFICATION– ANACONDA

Anaconda is an open-source package manager for Python and R. It is the most popular platform among data science professionals for running Python and R implementations. There are over three hundred libraries in data science, so having a robust distribution system for them is necessary for any professional in this field. Anaconda simplifies package deployment and management.

On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda, you can easily set up, manage, and share Cond environments. Moreover, you can deploy any required project with a fewclicks when you are using Anaconda. There are many advantages to using Anaconda and the following are the most prominent ones among them: Anaconda is free and open- source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda for Python because every recruiter.

It has more than 1500 Python and R data science packages, so you don't face any compatibility issues while collaborating with others. For example, suppose your colleague sends you a project which requires packages called A and B but you only have package A.

Without having package B, you wouldn't be able to run the project. Anaconda mitigates the chances of such errors. You can easily collaborate on projects without worrying about any compatibility issues. It gives you a seamless environment that simplifies deploying projects. You can deploy any project with just a few clicks and commands while managing the rest. Anaconda has a thriving community of data scientists and machine learning professionals who use it regularly. If you encounter an issue, chances are, the community has already answered the same. On the other hand, you can also ask people in the community about the issues you face there, it's a very helpful community ready to help new learners.

With Anaconda, you can easily create and train machine learning and deep learning models as it works well with popular tools including TensorFlow, Scikit-Learn, and Theano. You can create visualizations by using Bokeh, Holoviews, Matplotlib, and Datashader while using Anaconda.

3.4.1 How to Use Anaconda for Python

Now that we have discussed all the basics in our Python Anaconda tutorial, let's discuss some fundamental commands you can use to start using this package manager. Listing All Environments To begin using Anaconda, you'd need to see how many Conda environments are present in your machine. `conda env list` It will list all the available Conda environments in your machine. Creating a New Environment, You can create a new Conda environment by going to the required directory and use this command:

```
conda create -n <your_environment_name>
```

You can replace `<your_environment_name>` with the name of your environment. After entering this command, conda will ask you if you want to proceed to which you should reply with y: proceed ([y])/n)?

On the other hand, if you want to create an environment with a particular version of Python, you should use the following command:

```
conda create -n <your_environment_name> python=3.6
```

Similarly, if you want to create an environment with a particular package, you can use the following command:

```
conda create -n <your_environment_name> pack_name
```

Here, you can replace `pack_name` with the name of the package you want to use.

If you have a `.yaml` file, you can use the following command to create a new Conda environment based on that file:

```
conda env create -n <your_environment_name> -f <file_name>.yaml
```

We have also discussed how you can export an existing Conda environment to a `.yaml` file later in this article.

3.4.2 Activating an Environment

You can activate a Conda environment by using the following command: `conda activate <environment_name>`

You should activate the environment before you start working on the same. Also, replace the term `<environment_name>` with the environment name you want to activate. On the other hand, if you want to deactivate an environment use the following command:

```
conda deactivate.
```

3.4.3 Installing Packages in an Environment

Now that you have an activated environment, you can install packages into it by using the following command : `conda install <pack_name>`

Replace the term `<pack_name>` with the name of the package you want to install in your Conda environment while using this command.

3.4.4 Updating Packages in an Environment

If you want to update the packages present in a particular Conda environment, you should use the following command:

```
conda update
```

The above command will update all the packages present in the environment. However, if you want to update a package to a certain version, you will need to use the following command:

```
conda install <package_name>=<version>
```

3.4.5 Exporting an Environment Configuration

Suppose you want to share your project with someone else (colleague, friend, etc.). While you can share the directory on Github, it would have many Python packages, making the transfer process very challenging. Instead of that, you can create an environment configuration .yml file and share it with that person. Now, they can create an environment like your one by using the .yml file. For exporting the environment to the .yml file, you'll first have to activate the same and run the following command:

```
conda env export ><file_name>.yml
```

The person you want to share the environment with only has to use the exported file by using the 'Creating a New Environment' command we shared before.

3.4.6 Removing a Package from an Environment

If you want to uninstall a package from a specific Conda environment, use the following command

```
conda remove -n <env_name><package_name>
```

On the other hand, if you want to uninstall a package from an activated environment, you'd have to use the following command:

```
conda remove <package_name>
```

3.4.7 Deleting an Environment

Sometimes, you don't need to add a new environment but remove one. In such cases, you must know how to delete a Conda environment, which you can do by using the following command: `conda env remove --name <env_name>`

The above command would delete the Conda environment right away. Conda is a popular package management system used for managing dependencies and environments in Python. While creating a new environment in Conda is a common task, there may come a time when you need to delete an existing environment. This could be because the environment is no longer needed or because it's causing conflicts with other packages or environments on your system.

Here, `<env_name>` is the name of the environment you want to delete. When you run this command, Conda will immediately delete the specified environment and all its associated packages. It's important to note that once you delete an environment, it cannot be recovered. If you want to keep a backup of the environment, you should export its specifications to a YAML file before deleting it. It has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you are using Anaconda. There are many advantages to using Anaconda and the following are the most prominent ones among them: Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open source too, which has made it widely popular.

CHAPTER- 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

This diagram is nothing but a simple description of all the entities that have been incorporated into the system. The diagram represents the relations between each of them and involves a sequence of decision-making processes and steps. You can simply call it a visual or the whole process and its implementation. All functional correspondences are explained in this diagram.

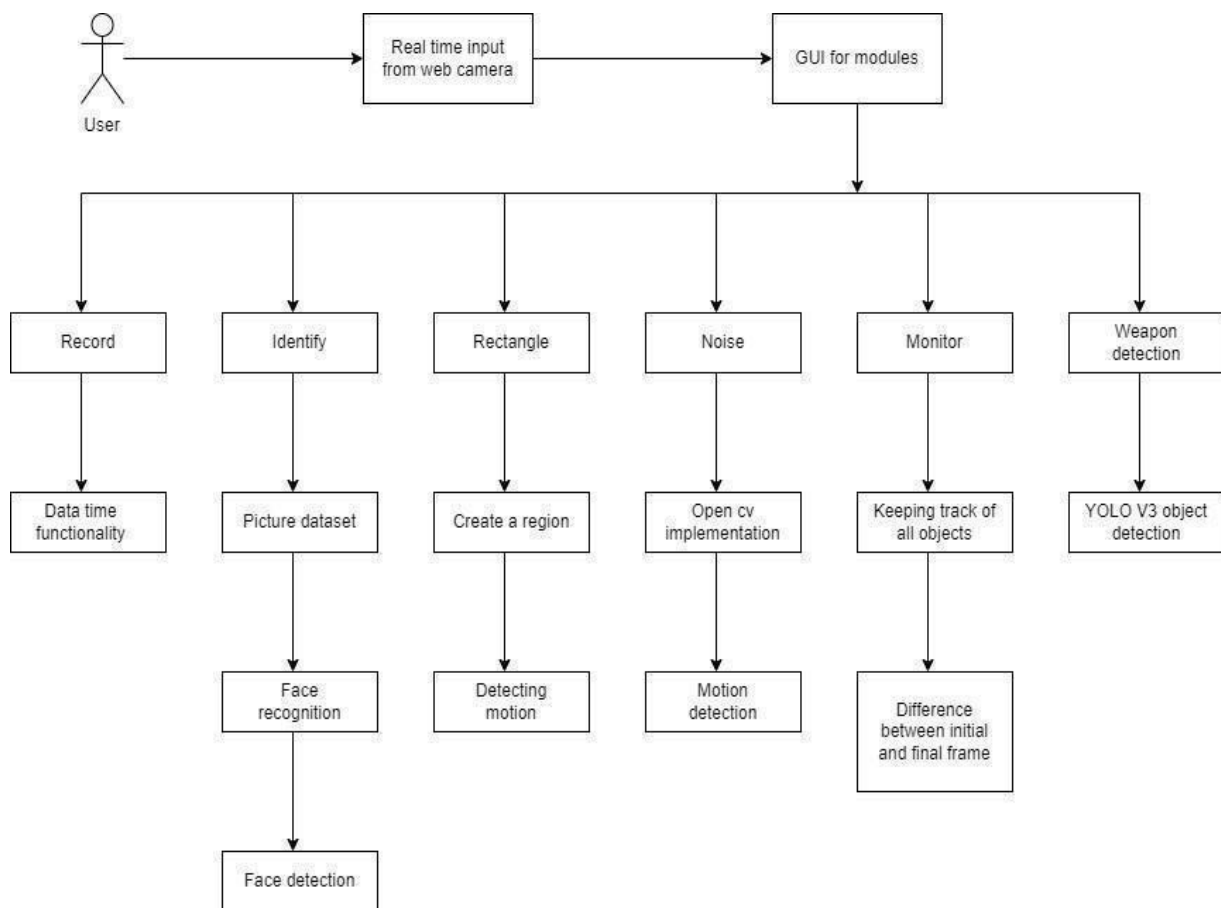


Fig 4.1 – Architecture Diagram

4.2 USE-CASE DIAGRAM

A use case diagram is a standard diagram that shows all interactions between the user, dataset, and algorithm used. It is developed in the early stages of the process.

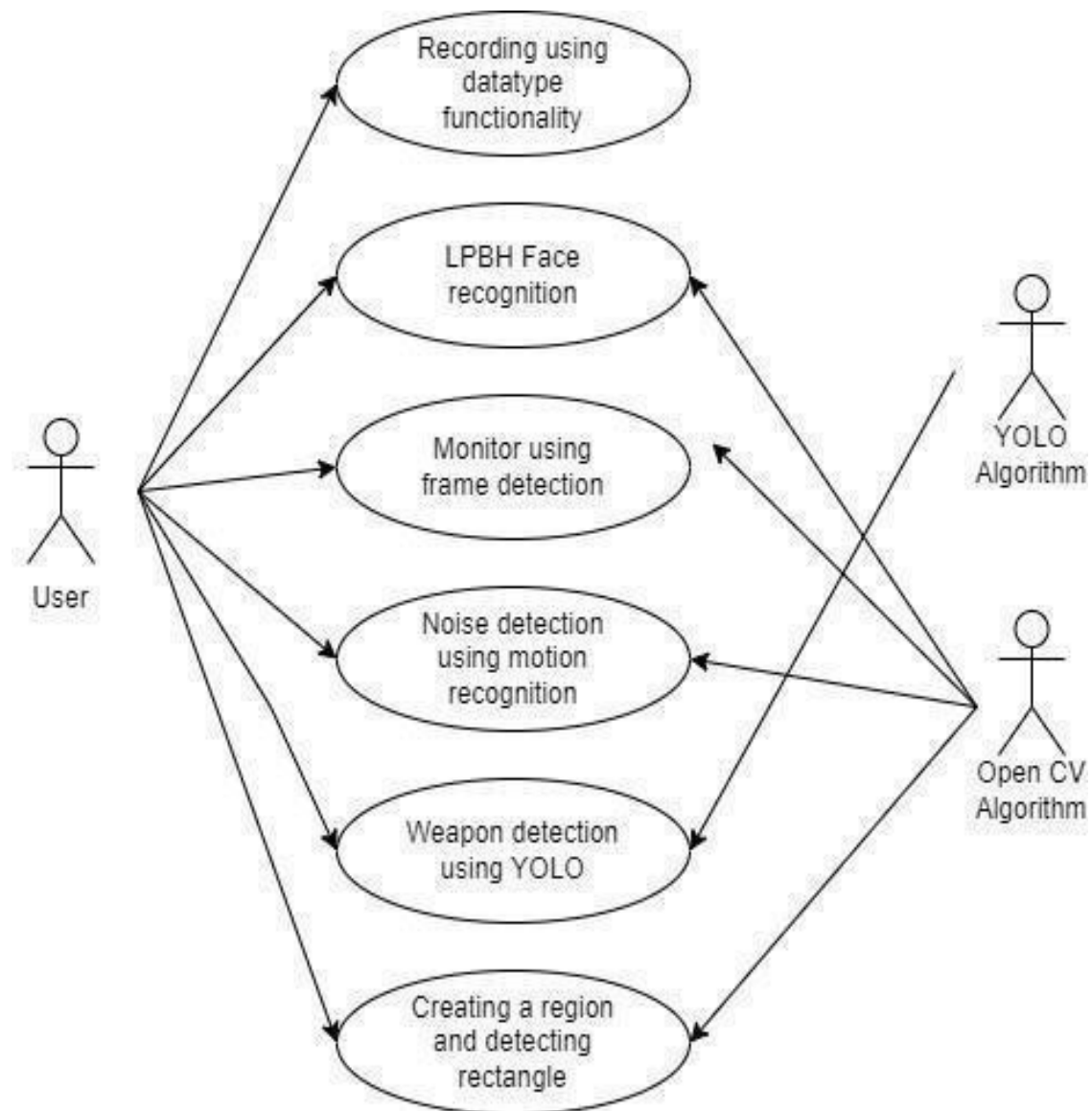


Fig 4.2 – Use-Case Diagram

4.3 ACTIVITY DIAGRAM

In simple terms, a diagram that represents the order of all activities is called the activity diagram. It shows the workflow between different activities that take place in the whole process. However, these are not exactly flowcharts but are similar.

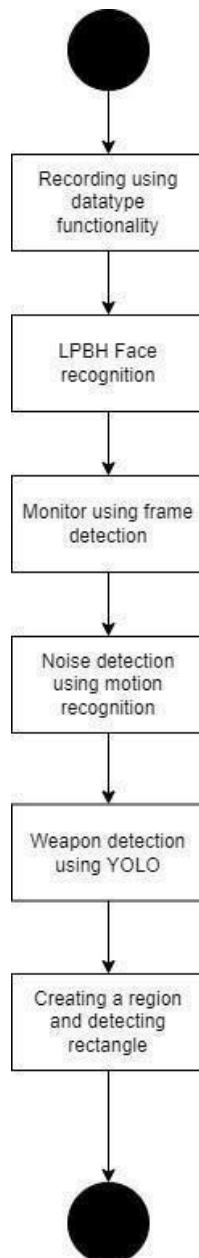


Fig 4.3 – Activity Diagram

4.4 DATA FLOW DIAGRAM LEVEL ZERO

This is basically a contextual diagram, also referred to as a “context diagram.” It only represents the top level or the 0 Level in the whole process. It gives an abstraction kind of view and shows the whole system as a single process and its relationship to externalities.

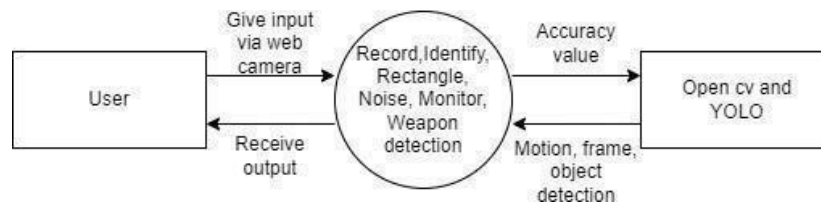


Fig 4.4 – Data Flow Diagram

4.5 COLLABORATION DIAGRAM

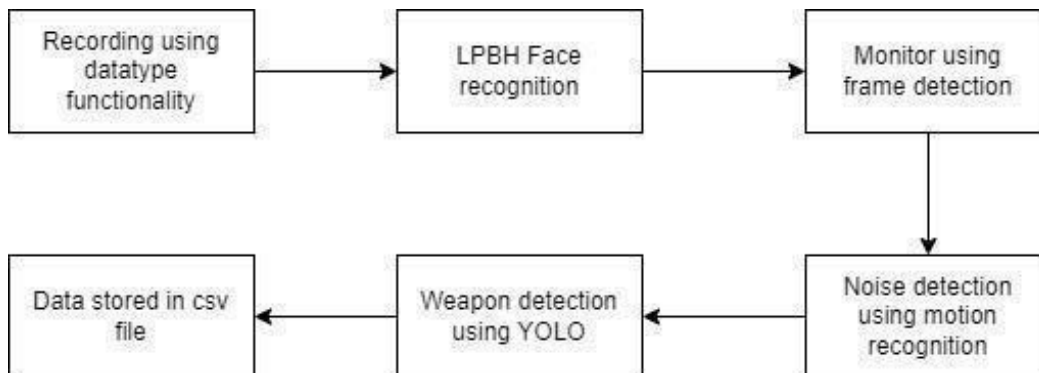


Fig 4.5 – Collaboration Diagram

4.5.1 Collaboration Diagram Explanation: -

Advanced video surveillance systems using computer vision have become increasingly popular in recent years due to their ability to provide real-time monitoring and detection of security threats. These systems use sophisticated machine learning algorithms and computer vision techniques to analyze video feeds from cameras installed in the surveillance area and identify objects of interest, such as people, vehicles, or suspicious behavior. Alerts are generated based on the objects detected, which can be sent to security personnel for immediate action. Additionally, the system can store video footage and other data

generated.

4.6 CLASS DIAGRAM

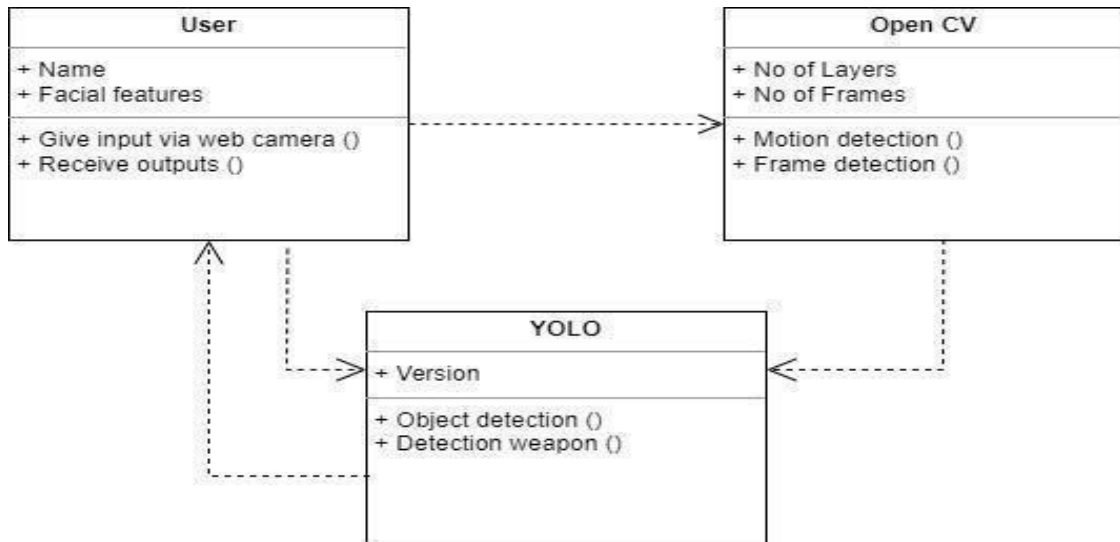


Fig 4.6 – Class Diagram

Computer Vision Class: This class represents the computer vision component that uses machine learning algorithms and computer vision techniques to analyze the video feeds and detect objects of interest, such as people, vehicles, or suspicious behavior.

Object Detection Class: This class represents the object detection component that receives data from the computer vision component and identifies the objects detected in the video feeds.

Alert Generation Class: This class represents the alert generation component that generates alerts based on the objects detected in the video feeds.

Storage Class: This class represents the storage component that stores the video footage and other data generated by the system, such as alerts and object detection data.

Security Personnel Class: This class represents the security personnel component that receives alerts generated by the system and takes appropriate action, such as investigating the incident or dispatching security personnel to the location.

CHAPTER - 5

MODULE DESCRIPTION

LIST OF MODULES:

- Monitor
- Identify
- Using LBPH for Face Recognition
- Noise Detection
- Visitors in Room Detection
- Weapon Detection

5.1 MONITOR:

This feature is used to find what is the thing which is stolen from the frame which is visible to the webcam. It constantly monitors the frames and checks which object or thing from the frame has been taken away by the thief. This uses Structural Similarity to find the differences in the two frames. The two frames are captured first when noise did not happen and second when the noise stopped happening in the frame. SSIM issued as a metric to measure the similarity between two given images. The Structural Similarity Index (SSIM) metric extracts 3 key features from an image:

- Luminance
- Contrast
- Structure

5.2 IDENTIFY

Face detection is done via Haarcascade classifiers which are again in-built in the OpenCV module of python. Cascade classifier, or namely cascade of boosted classifiers working with haar-like features, is a special case of ensemble learning, called boosting. It typically relies on Adaboost classifiers. Cascade classifiers are trained on a few hundred sample images of the image that contain the object we want to detect, and other images that do not contain those images. Inbuilt function called cascade classifier which we have used in order to detect faces in the frame. The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by

highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameter's radius and neighbors. Using the image generated in the last step, we can use the Grid X and Grid Y parameters to divide the image into multiple grids. The model is trained and later on when we want to make predictions the same steps are applied to the make and its histograms are compared with the already trained model. To train a machine learning model using a grid-based approach, you would first need to collect a set of data that includes examples of the object or phenomenon you want to recognize. This could be images, audio recordings, or other types of data. You would then need to preprocess this data to prepare it for use in the model.

This might involve resizing or scaling images, filtering audio recordings, or other steps depending on the nature of the data. Once the data has been preprocessed, you can begin extracting features from it. In a grid-based approach, you might divide an image into multiple grids using the Grid X and Grid Y parameters you mentioned. For each grid, you could calculate a set of features that represent the content of that grid. For example, you might calculate histograms of color values, edge orientations, or other image features.

Once you have extracted features from your data, you can use these features to train a machine learning model. Depending on the specifics of your approach, you might use a supervised learning algorithm like a support vector machine or a deep neural network. You would typically provide the model with labeled examples of the object or phenomenon you want to recognize, and the model would learn to associate the extracted features with the corresponding labels. After training the model, you can use it to make predictions on new data. To do this, you would apply the same feature extraction process to the new data, dividing it into grids and calculating feature histograms for each grid. You would then compare these feature histograms to the ones used to train the model, using a distance metric like Euclidean distance or cosine similarity. The prediction would be based on the closest matching feature histograms.

Overall, a grid-based approach to feature extraction and comparison can be effective for certain types of machine learning problems. However, it is not the only

approach and may not be the best choice for all situation

5.3 RECTANGLE:

In this module, initially an rectangle box is created and then the rectangle box will be monitored. A region will be created inside the rectangle box and each frames will be detected and then the motion is predicted inside the rectangle box only.

5.4 NOISE DETECTION:

This feature is used to find the noises in the frames. This is something found in most CCTV but in this module, we'll see how it works. All the frames are continuously analyzed and checked for noises. Noise is checked in the consecutive frames. Simply we do the absolute difference between two frames and in this way the difference between two images is analyzed and Contours(boundaries of the motion are detected) .If there are no boundaries then no motion and if there is any there is motion. All images are just integer/ float values of pixels which tells the brightness of pixel and similarly every pixel has values of brightness. So we just do simply absolute difference because negative will make no sense at all.

5.5 RECORD:

In this module, When the person turns on the web camera, real time video feed is captured, and the feed is recorded using the date time functionality. This is done using the Open CV Architecture where the time is exactly captured, and the video feed is recorded. The OpenCV (Open Source Computer Vision) library is a popular open-source computer vision and machine learning software library that is used to develop real-time computer vision applications. It provides a range of tools and functions that can be used to process and analyze images and videos. In the context of surveillance systems, OpenCV can be used to capture real-time video feed from a web camera and record the feed using the date-time functionality. This involves using the library's functions to capture frames from the video stream and write them to a file with a filename that includes the current date and time.

OpenCV provides several functions for capturing video feed from web cameras, including the "Video Capture" function which allows for the capture of frames from a video stream. The "inwrite" function can be used to write the captured frames to a file with a specific filename that includes the date and time. By combining these functions, a real-time surveillance system can be developed that captures video feed and records it with the date and time for future reference. This can be useful in identifying specific events or incidents that occur in the area being monitored and can provide a record of activity over time. However, it is important to note that the use of surveillance systems must be done in accordance with applicable laws and regulations to ensure the privacy and security of individuals in the area being monitored. By combining these functions, a real-time surveillance system can be developed that captures video feed and records it with the date and time for future reference. This can be useful in a variety of applications, such as security systems, traffic monitoring, and wildlife observation.

You can also add additional functionality to this system, such as motion detection, object tracking, and facial recognition, by using OpenCV's built-in algorithms and libraries. This can help you to further enhance the surveillance capabilities of your system and make it more effective in detecting and responding to potential threats or incidents. OpenCV is a popular open-source computer vision library that provides a range of pre-built algorithms and tools for developing applications related to image and video processing, object detection and tracking, facial recognition, and other computer vision tasks.

By integrating OpenCV into a surveillance system, you can enhance its capabilities with advanced functionality like motion detection, object tracking, and facial recognition. For example, you could use OpenCV's background subtraction algorithm to detect changes in an image or video stream, allowing you to identify movement and potential threats in real-time.

Object tracking is another feature that can be added to a surveillance system using OpenCV. You can use algorithms like mean shift or CamShift to track objects or people in a video stream, allowing you to monitor their movements and

behavior over time.

5.6 WEAPON DETECTION:

The final step is the detection of weapon from the real time web camera. Once the system receives an input, it goes through the VGG19 layers and is boosted by Transfer Learning to predict the final output. Following which, we import necessary libraries, and an API Path is created for the same. The Proposed system has a feature where the weapon is predicted from the real time video. The final step in the proposed system for weapon detection involves the use of real-time video from a web camera to predict the presence of a weapon. The system takes the input from the camera and processes it using the VGG19 neural network. The VGG19 network is a deep convolutional neural network that is commonly used for image classification tasks. In this case, the network is used to detect the presence of a weapon in the real-time video stream.

To improve the accuracy of the weapon detection, Transfer Learning is used to boost the prediction of the VGG19 network. Transfer Learning is a machine learning technique where a pre-trained model is used as a starting point for a new task. In this case, the pre-trained VGG19 model is used as the starting point for detecting weapons in the real-time video stream. After processing the real-time video stream, the system predicts the presence of a weapon in the video. To implement this, the necessary libraries are imported, and an API Path is created for the same. The API Path is used to transfer the input video from the camera to the system for processing and prediction.

Overall, the proposed system provides a reliable and efficient method for detecting weapons in real-time video streams. By using deep learning techniques like Transfer Learning and the VGG19 network, the system is able to accurately predict the presence of weapons, which can be crucial in ensuring.

5.7 Functional Tests:

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files. Functional test cases are designed to test the functionality of the software system and ensure that it meets the specified requirements. One aspect of functional testing is testing the code with nominal input values, which are inputs that are within the normal range of values that the software system is expected to handle.

In addition to nominal input values, it's important to also test the software system with boundary values, which are inputs that are at the edge of the range of values that the system is expected to handle. This includes testing the software system with the minimum and maximum values, as well as values that are just above or below the minimum and maximum values. Testing boundary values is important because it helps to ensure that the software system can handle inputs that are close to the limits of what it can handle.

Special values are also important to test, such as logically related inputs, files of identical elements, and empty files. Logically related inputs refer to inputs that are related in some way, such as inputs that are part of the same calculation or inputs that are related to each other in some other way. Testing with logically related inputs helps to ensure that the software system can handle complex calculations and relationships between inputs.

Files of identical elements refer to input files that contain the same value for every element. Testing with these types of files helps to ensure that the software system can handle large amounts of data and can process data efficiently. To implement this, the necessary libraries are imported, and an API Path is created for the same.

There are Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

5.8 Performance Test

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

5.9 Stress Test

Stress Test is a test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

5.10 Structure Test

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.

Handling end of file conditions, I/O errors, buffer problems, and textual errors in the output information

5.11 Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules that makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should

be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then test. This approach evolved from the unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that interface discrepancies can be easily found and corrected.

The major error that was faced during the project is a linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated as they complete unit testing. Testing is completed when the last module is integrated and tested.

Based on the information you provided, it seems that you are talking about a modular approach to software development, where a complex system is broken down into smaller, more manageable modules that can be developed, tested, and integrated independently. This approach has several advantages, including easier debugging and maintenance, better scalability, and faster development times. One of the challenges that you mentioned is linking errors, which occur when there is a problem with the connections between different modules. This can be especially challenging when working with large and complex systems. To avoid this, it's important to carefully plan and document the interfaces between different modules and to test them thoroughly before integrating them into the larger system.

Another challenge that can arise when using a modular approach is ensuring that all the modules work together seamlessly. This can be addressed by conducting thorough unit testing on each module before integrating them into the larger system. By doing so, you can catch any issues early on and address them before they become more difficult.

6.1 TESTING TECHNIQUES / TESTING STRATEGIES

Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet-undiscovered error. A successful test is one that uncovers an as-yet-undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hangs together. System testing requires a test consisting of several key activities and steps for running a program, string, and system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise, the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that has a probability of finding a yet undiscovered error. A successful test is one that uncovers a yet undiscovered error. Any engineering product can be tested in one of the two ways:

6.2 White box testing

This testing is also called Glass box testing. In this testing, by knowing the specific functions that a product has been designed to perform tests can be conducted that demonstrate each function is fully operational and at the same time search for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is white box testing.

Basis path testing:

- Flow graph notation
- Cyclomatic complexity
- Deriving test cases
- Graph matrices Control

6.2 Black box testing

In this testing by knowing the internal operation of a product, a test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black-box test case design are:

- Graph-based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

6.3 SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. For this reason, a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer-based system.
- Different testing techniques are appropriate at different points in time.
- Testing and Debugging are different activities but debugging must be

accommodated in any testing strategy.

6.4 Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with it. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem, of course, is “putting them together”- interfacing. There may be the chances of data loss across another’s sub- functions when combined may not produce the desired major function; individually acceptable impressions may be magnified to unacceptable levels; global data structures can present problems.

6.5 Program Testing

The logical and syntax errors have been pointed out by program testing. Asyntax error is an error in a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax errors. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-of-range items, and invalid combinations. Since the compiler will not deduct logical errors, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or an arithmetic expression. The condition testing method focuses on testing each condition in the program. The purpose of the condition test is to deduct not only errors in the condition of a program but also other errors in the program. a condition is a statement that evaluates to either true or false, and is typically used to control the flow of execution in a program. A condition can include a variety of elements, including Boolean operators (such as AND, OR, and NOT), Boolean variables (which are variables that can only take on the values true or false), parentheses (which can be used to group elements together), relational operators (such as equals, less than, or greater than), and arithmetic expressions (which can be used to compare values).

6.6 Security Testing

Security testing attempts to verify the protection mechanisms built into a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attacks must also be tested for invulnerability from rear attacks. During security, the tester places the role of the individual who desires to penetrate the system. During security testing, the tester simulates various attack scenarios that a potential attacker might use to gain unauthorized access to the system. These scenarios can include brute-force attacks, password cracking, SQL injection attacks, and other forms of cyberattacks. By simulating these attacks, the tester can identify vulnerabilities and weaknesses in the system's security mechanisms and recommend appropriate remedial actions to enhance the system's security.

The security testing process can be performed using various techniques, including vulnerability scanning, penetration testing, and ethical hacking. Vulnerability scanning involves using automated tools to scan the system for known vulnerabilities and security weaknesses, while penetration testing involves manual testing to identify and exploit vulnerabilities that may not be detected by automated tools. Ethical hacking involves simulating real-world attacks to test the system's resilience to such attacks and identify areas that require improvement. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda, you can easily set up, manage, and share Conda environments.

Moreover, you can deploy any required project with a few clicks when you are using Anaconda. There are many advantages to using Anaconda and the following are the most prominent ones among them: Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda.

6.7 Validation Testing

At the culmination of integration testing, the software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that is reasonably expected by the customer. Software validation is achieved through a series of black-box tests that demonstrate conformity with requirements. After the validation test has been conducted, one of two conditions exists.

- * The function or performance characteristics conform to specifications and are accepted.
- * A validation from the specification is uncovered and a deficiency is created. Deviations or errors discovered at this step in this project are corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies.

Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic

6.8 User Acceptance Testing

User acceptance of the system is a key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective systems and users at the time of developing and making changes whenever required. This is done in regard to the following points.

- Input screen design.
- Output screen design.

CHAPTER – 7

RESULTS AND DISCUSSIONS

In this project, algorithms are proposed by Sum of Absolute Difference for recording videos only when motion is detected. The major advantage of this algorithm is that less storage is required as the system records only for frames with a particular threshold. Even the amount of processing required is reduced. The system efficiency is calculated to be around 72%. As recognized images get stored with the name of the person, there is ease of access for surveillance security personnel. Thus, the implementation can be done in industrial spaces or places of high security like nuclear power plants, where entry is restricted to authorized personnel only. Further, the system can be implemented in farm or forest surveillance where detection or spotting of a specific animal can be recorded and on recognizing potential threat an alarm can be triggered.

The main objective of this project is to develop an algorithm that is able to detect and track moving objects. In this thesis we presented a set of methods and tools for moving object detection and tracking system. We implemented three different object detection algorithms and compared their detection quality and time-performance. Our thesis includes the following two main building blocks: Moving Object Detection and Object Tracking. Moving Object detection segments the moving targets from the background, and it is the crucial first step in video surveillance. In this thesis we implemented and tested temporal differencing object detection algorithms, background subtraction, and Hybrid adaptive scheme algorithms, and compared their detection quality and time performance.

OUTPUT SCREENSHOTS

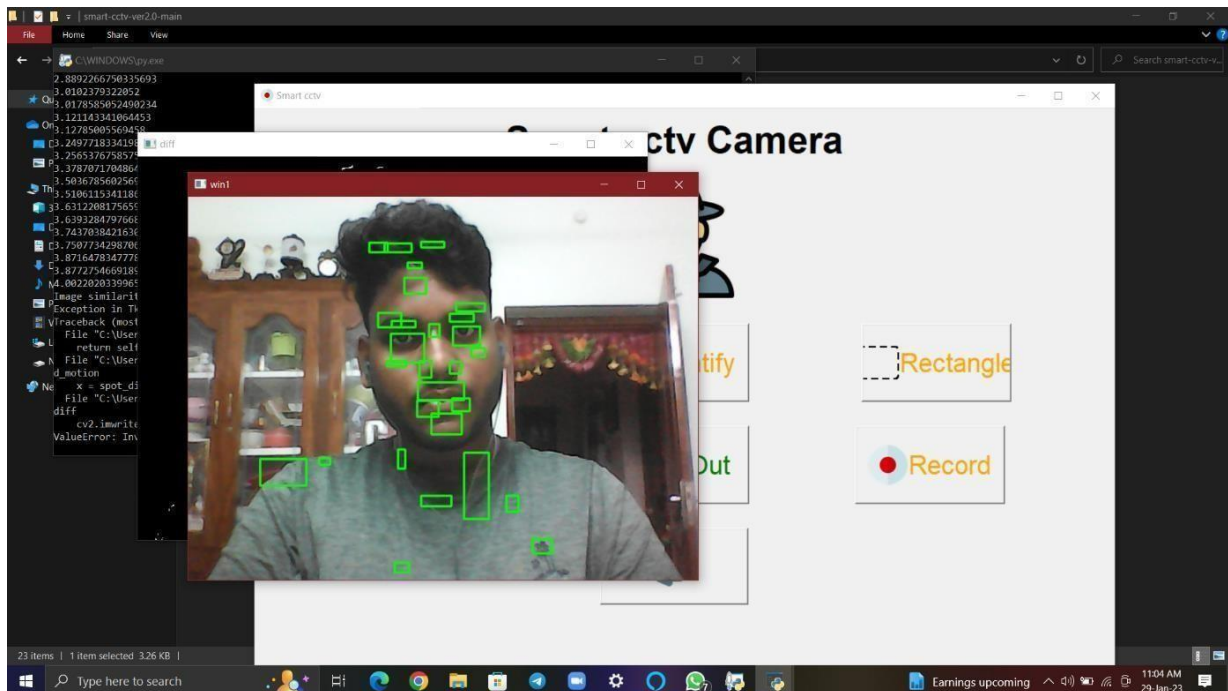


Fig 7.1 Identifying name of the person

Explanation:-

To identify the name of a person, the system can be trained using a database of images and names of known individuals. When the system detects a face in a video feed, it can compare the facial signature of the detected face with the database of known individuals to find a match. Once a match is found, the system can display the name of the person associated with the matching facial signature.

However, it is important to note that facial recognition technology has raised concerns regarding privacy and civil liberties, and its use should be carefully regulated and monitored to prevent misuse or abuse. In some jurisdictions, there are regulations that govern the use of facial recognition technology, and it is important to ensure that the system is compliant with applicable laws and regulations.

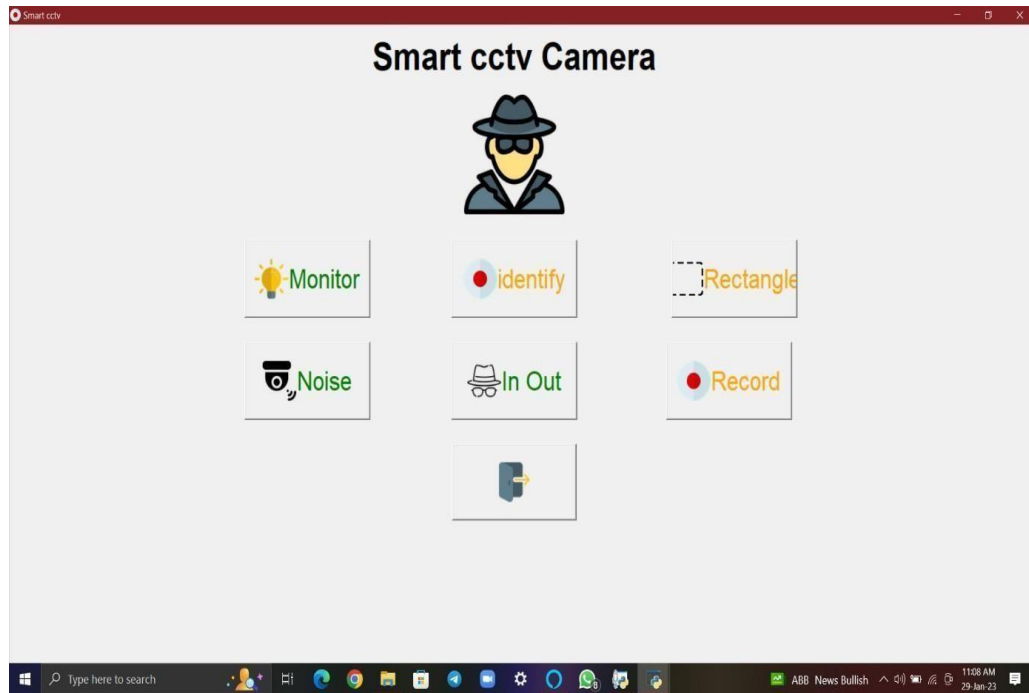


Fig 7.2 Main GUI Window

Explanation:-

The main GUI (**Graphical User Interface**) window of an advanced video surveillance system using computer vision is the primary interface through which the user interacts with the system. It provides a visual representation of the surveillance area and the video feeds received from the cameras installed in the area.

The main GUI window typically includes the following elements:

Video Feed Display: This element displays the video feeds received from the cameras installed in the surveillance area. The video feed can be displayed in real-time or as recorded footage.

Object Detection Display: This element displays the objects detected by the computer vision component of the system. The objects can be highlighted in the video feed or displayed in a separate window.

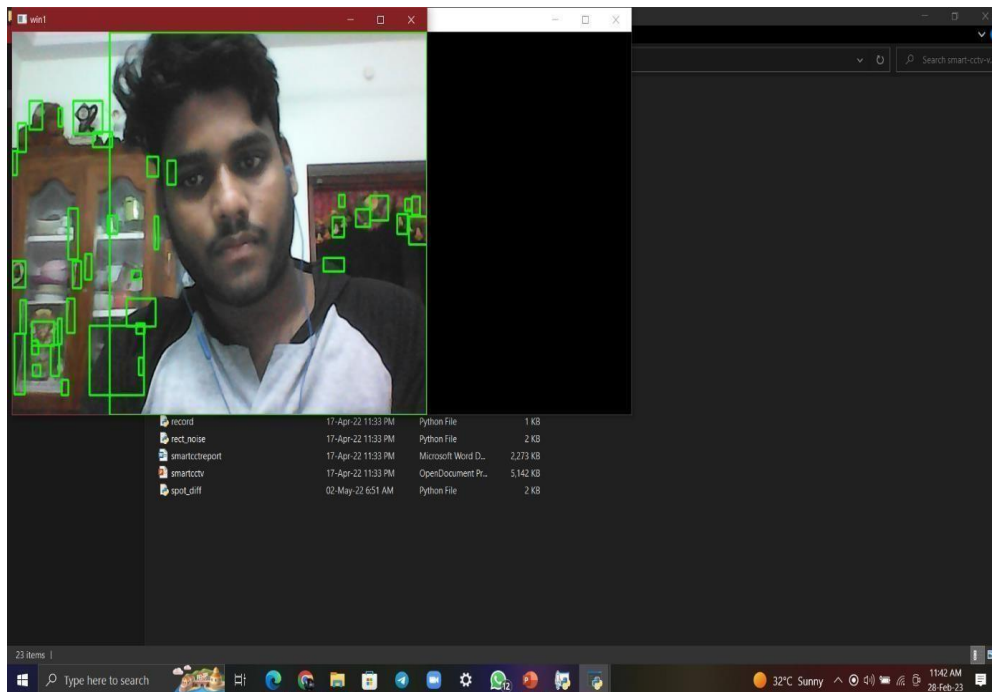


Fig 7.3 Unknown identification

Explanation:-

Unknown identification is typically performed using object detection algorithms that are trained to detect objects of interest in the video feeds. The system can be configured to generate alerts when objects that meet certain criteria are detected. For example, the system can be configured to generate an alert when an unknown person enters a restricted area or when a vehicle that does not belong to any authorized personnel is detected in the surveillance area.

To perform unknown identification, the system can use a combination of computer vision techniques such as object detection, object tracking, and facial recognition. Object detection algorithms can be used to detect objects of interest in the video feeds, while object tracking algorithms can be used to track the detected objects across different video frames. Facial recognition algorithms can be used to match the detected faces with known faces in the system database, or to generate a facial signature for the unknown person that can be used for future identification.

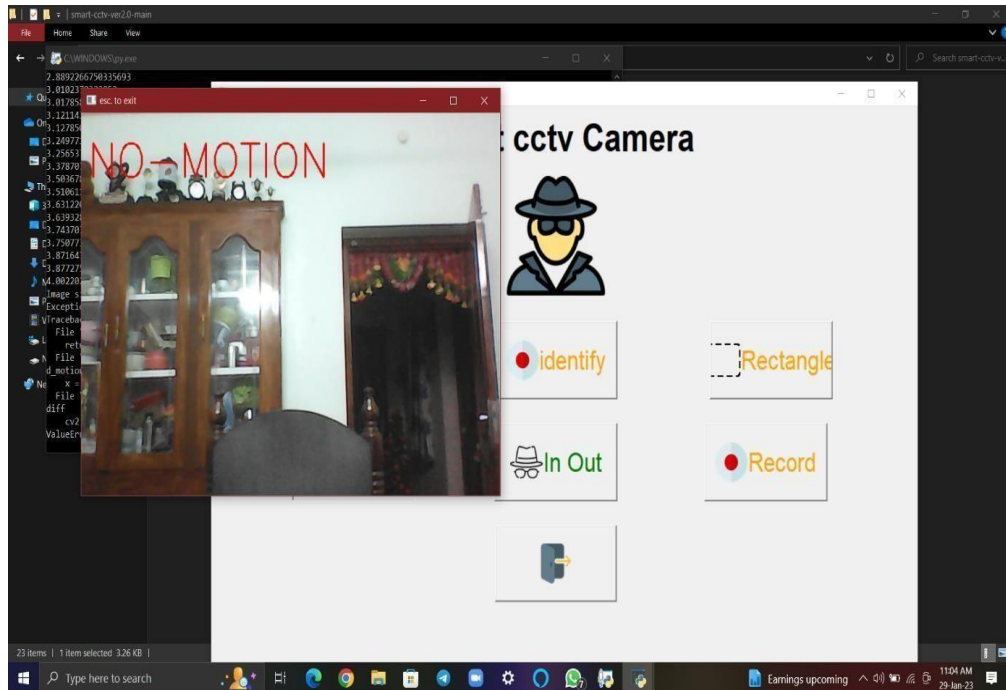


Fig 7.4 No motion detection

Explanation:-

To perform no motion detection, the system can analyze the video feed from the camera using computer vision algorithms. The system can be programmed to identify and track objects or people in the surveillance zone and to detect when there is no movement detected in a particular area or zone. When no movement is detected in a zone, the system can generate an alert or notification, indicating that there is a potential security risk or a potential malfunction in the system.

However, it is important to note that no motion detection can have limitations, such as false alarms caused by environmental factors such as wind, shadows, or changes in lighting conditions. Therefore, it is essential to configure the system properly and set appropriate thresholds for no motion detection to avoid false alarms and ensure accurate detection.

The implementation of no motion detection in a surveillance system involves the use of computer vision algorithms to analyze video feed from a camera and identify and track objects or people in the surveillance zone. This can

be done using techniques such as background subtraction, optical flow, and object detection. Once the objects or people in the surveillance zone have been identified and tracked, the system can detect when there is no movement detected in a particular area or zone. This is typically done by comparing the current frame with the previous frame and checking for differences in the position of the tracked objects or people.

When no movement is detected in a zone, the system can generate an alert or notification, indicating that there is a potential security risk or a potential malfunction in the system. This can be particularly useful in detecting intrusion attempts, as well as equipment failures or malfunctions. Motion detection is a key feature in many security systems, and it can be used to detect potential intruders or other security threats. With the help of motion sensors and cameras, security systems can detect when there is movement in a specific area or zone, and alert the user or trigger an alarm. This can be useful in a variety of settings, from homes to businesses to public spaces. When a security system detects motion or other potential security threats, it can generate alerts or notifications to alert the user or a security team. These alerts can be sent via email, text message, or other means, and can provide information about the location and nature of the threat. With the help of alerts and notifications, security teams can respond quickly to potential security threats and take appropriate action to mitigate the risk. In addition to detecting security threats, motion detection can also be used to monitor equipment and detect malfunctions or other issues. By monitoring the movement of equipment, such as motors or conveyor belts, a system can detect when there is a problem and generate alerts or notifications to alert maintenance teams or operators. This can help prevent equipment failures and downtime and ensure that production processes run smoothly.

However, as mentioned earlier, no motion detection can have limitations and can be prone to false alarms caused by environmental factors such as wind, shadows, or changes in lighting conditions. To mitigate these issues, it is essential to configure the system properly and set appropriate thresholds for no motion detection.

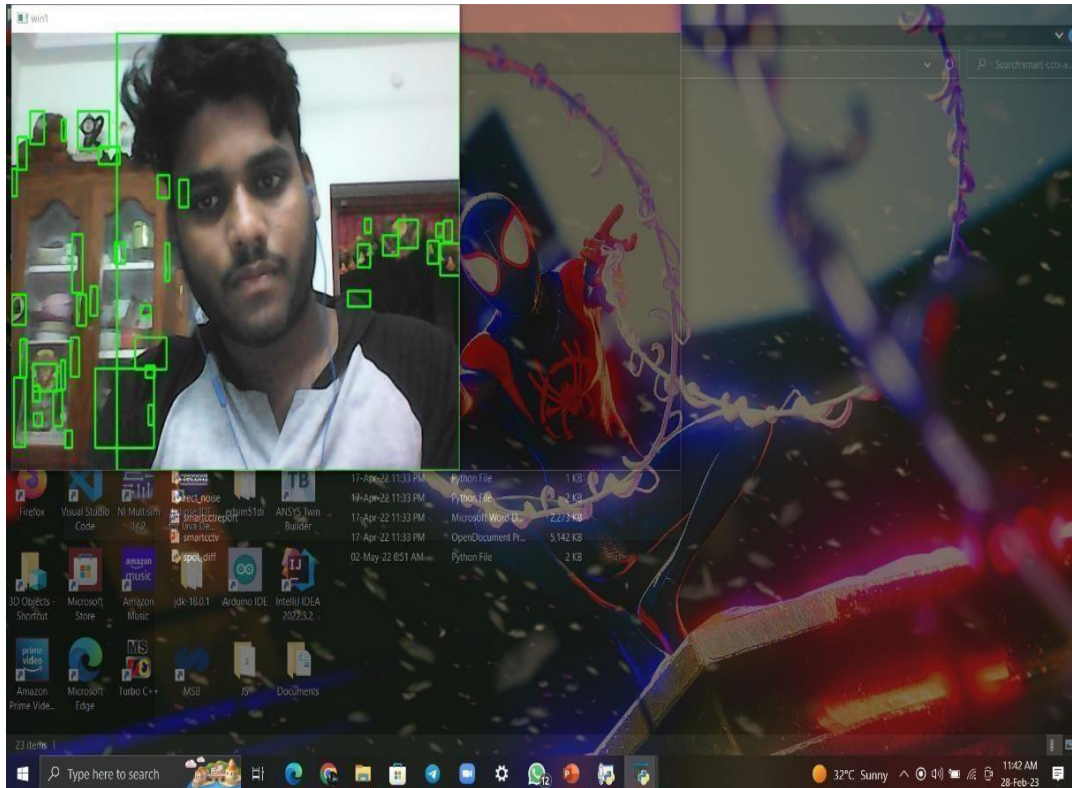


Fig 7.5 Motion detection

Explanation:-

To perform motion detection, the system can analyze the video feed from the camera using computer vision algorithms. The system can be programmed to identify and track objects or people in the surveillance zone and to detect any changes in their position or movement. When motion is detected in the zone, the system can generate an alert or notification, indicating a potential security risk or a potential event of interest. Motion detection can be useful in a variety of surveillance applications, such as detecting intruders or unauthorized access in a restricted area, monitoring for suspicious behavior or activities, or identifying potential safety hazards such as falling objects or hazardous spills.

To enhance the accuracy of motion detection, the system can be configured to eliminate false alarms by setting appropriate thresholds for motion detection, such as minimum size or speed of motion, and by filtering out environmental factors that may cause false alarms, such as shadows or changes in lighting conditions.

The system can also be programmed to track the detected motion across different video frames to provide additional information about the direction and speed of the movement.

Object recognition and classification: The system can use object recognition and classification algorithms to differentiate between different types of objects or people in the surveillance zone. This can help to identify potential threats more accurately and reduce the number of false alarms.

Integration with other sensors: The system can be integrated with other sensors, such as infrared sensors or sound sensors, to provide additional data for motion detection. This can improve the accuracy of the system and reduce false alarms.

Machine learning and AI: The system can be trained using machine learning and AI algorithms to learn and adapt to different environments and situations. This can help to improve the accuracy of the system over time and reduce the number of false alarms.

Multi-camera tracking: The system can be configured to track motion across multiple cameras in a surveillance network. This can provide a more comprehensive view of the surveillance zone and improve the accuracy of motion detection.

Privacy considerations: The system must be designed to protect the privacy of individuals in the surveillance zone and comply with relevant data protection regulations. This can include using encryption, access controls, and secure data storage, as well as implementing privacy policies and procedures. Surveillance systems can raise concerns about privacy and data protection, especially when they are used in public spaces or other areas where individuals may not expect to be monitored. To address these concerns, surveillance systems must be designed to protect the privacy of individuals in the surveillance zone and comply with relevant data protection regulations. This can include using encryption and other security measures to protect data in transit and at rest, as well as implementing access controls and secure data storage to prevent unauthorized access.



Fig 7.6 Rectangle - Motion detection

Explanation:-

To perform rectangle motion detection, the system can be configured to define a specific area or region of interest using a rectangular shape within the surveillance zone. The system can then analyze the video feed from the camera using computer vision algorithms to detect any movement within this specific rectangular region. When motion is detected in the defined region, the system can generate an alert or notification, indicating a potential security risk or a potential event of interest.

Rectangle motion detection can be useful in a variety of surveillance applications where only certain areas within the surveillance zone require monitoring. For example, in a parking lot, rectangle motion detection can be used to monitor only specific parking spots or entrances to detect any unauthorized access or suspicious activities. For example, object tracking algorithms can track the movement of objects or individuals across multiple cameras, allowing the system to build a more complete picture of the surveillance zone.

To enhance the accuracy of rectangle motion detection, the system can be configured to set appropriate thresholds for motion detection, such as minimum size or speed of motion, and by filtering out environmental factors that may cause false alarms, such as shadows or changes in lighting conditions.

Threat detection: An intelligent surveillance system can be designed to identify potential threats within the surveillance zone, such as suspicious behavior, unusual activity, or the presence of unauthorized individuals or objects. By analyzing video feeds and integrating with other sensors and systems, the system can provide real-time alerts and notifications to security personnel and emergency services, enabling them to respond quickly and effectively to potential security threats.

Crowd management: In crowded public areas, such as airports, train stations, or sports stadiums, an intelligent surveillance system can be used to manage crowd movement and prevent overcrowding. The system can monitor the number of people in a particular area and provide real-time information to crowd management personnel, allowing them to adjust staffing levels, redirect foot traffic, or implement other measures to maintain crowd control and ensure public safety.

Traffic management: An intelligent surveillance system can be used to manage traffic flow and reduce congestion on roads and highways. By integrating with traffic sensors, GPS systems, and other data sources, the system can provide real-time traffic updates and route recommendations to drivers, allowing them to avoid congestion and reach their destinations more quickly and safely.

Environmental monitoring: An intelligent surveillance system can be used to monitor environmental conditions, such as air quality, water quality, and weather patterns. By integrating with sensors and other data sources, the system can provide real-time updates and alerts to environmental agencies and emergency services, enabling them to respond quickly to environmental threats and minimize the impact on public health and safety.

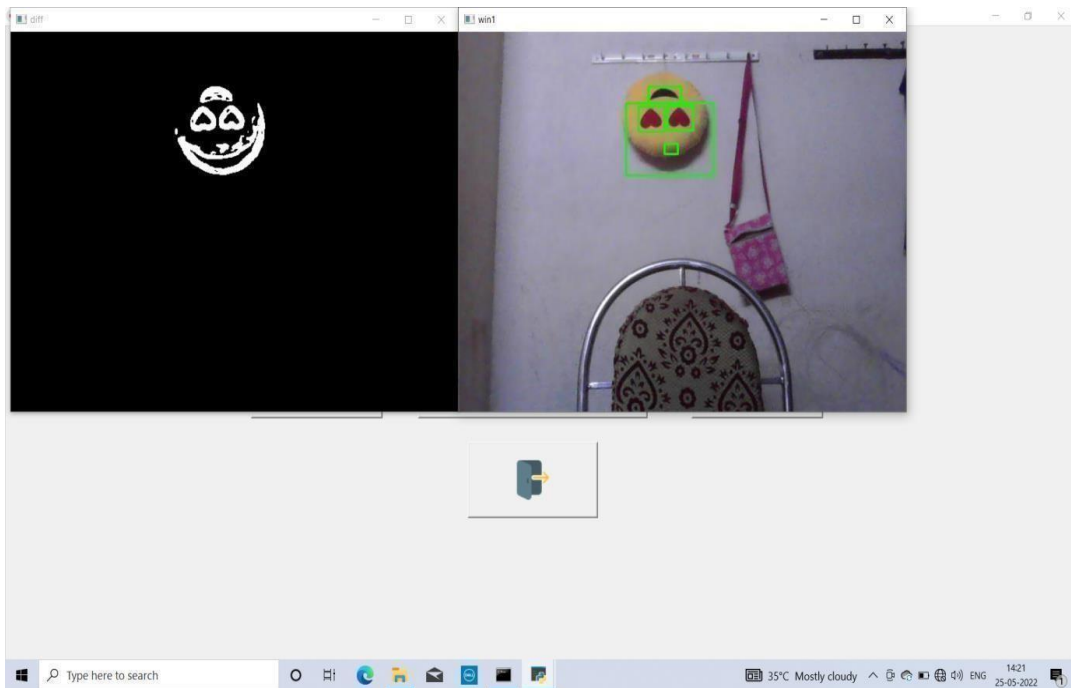


Fig 7.7 Spot the difference detection

Explanation:-

To perform spot the difference detection, the system can be configured to analyze consecutive video frames from the camera using computer vision algorithms. The system can then compare the current frame with the previous frame to detect any changes or differences in the video feed. When a change or difference is detected, the system can generate an alert or notification, indicating a potential security risk or a potential event of interest.

Spot the difference detection can be useful in a variety of surveillance applications where subtle changes in the surveillance zone need to be monitored. For example, in a bank or a casino, spot the difference detection can be used to monitor for suspicious behavior, such as someone removing or adding objects from a table or a cash register.

To enhance the accuracy of spot the difference detection, the system can be programmed to filter out environmental factors that may cause false alarms, such as changes in lighting conditions or camera movement. The system can also be configured to set appropriate thresholds for detecting changes or differences.

In addition to generating alerts or notifications, spot the difference detection can also be used for post-event analysis and investigation. By analyzing the video feed before and after an event, the system can provide additional information and context about the event, such as the time and location of the event, the individuals involved, and the direction and speed of movement.

Spot the difference detection can also be integrated with other sensors and systems to provide a more comprehensive surveillance solution. For example, it can be combined with motion detection to provide additional information about the nature and direction of the detected motion, or with face recognition to identify the individuals involved in the detected change or difference. This integration can help to improve the accuracy and reliability of the surveillance system and enable more effective security response. Overall, a video surveillance system using computer vision can be a powerful tool in enhancing security and safety in a variety of settings, from public spaces to high-security facilities. With the ability to detect potential security threats in real-time, the system can help prevent security breaches and respond quickly to potential incidents, ultimately helping to protect individuals and assets.

In addition to detecting the presence of a weapon, the system can be programmed to track the movement of the detected object or person to provide additional information about the location and trajectory of the potential threat. This can be particularly useful in scenarios where multiple cameras are used to monitor a large area, as it can help security personnel quickly locate the source of the threat. Simply we do the absolute difference between two frames and in this way the difference between two images is analyzed and Contours(boundaries of the motion are detected) .If there are no boundaries then no motion and if there is any there is motion. All images are just integer/ float values of pixels which tells the brightness of pixel and similarly every pixel has values of brightness. So we just do simply absolute difference because negative will make no sense at all. Computer Vision is an interdisciplinary field of study that enables machines to interpret and understand digital images and videos from the world around us.

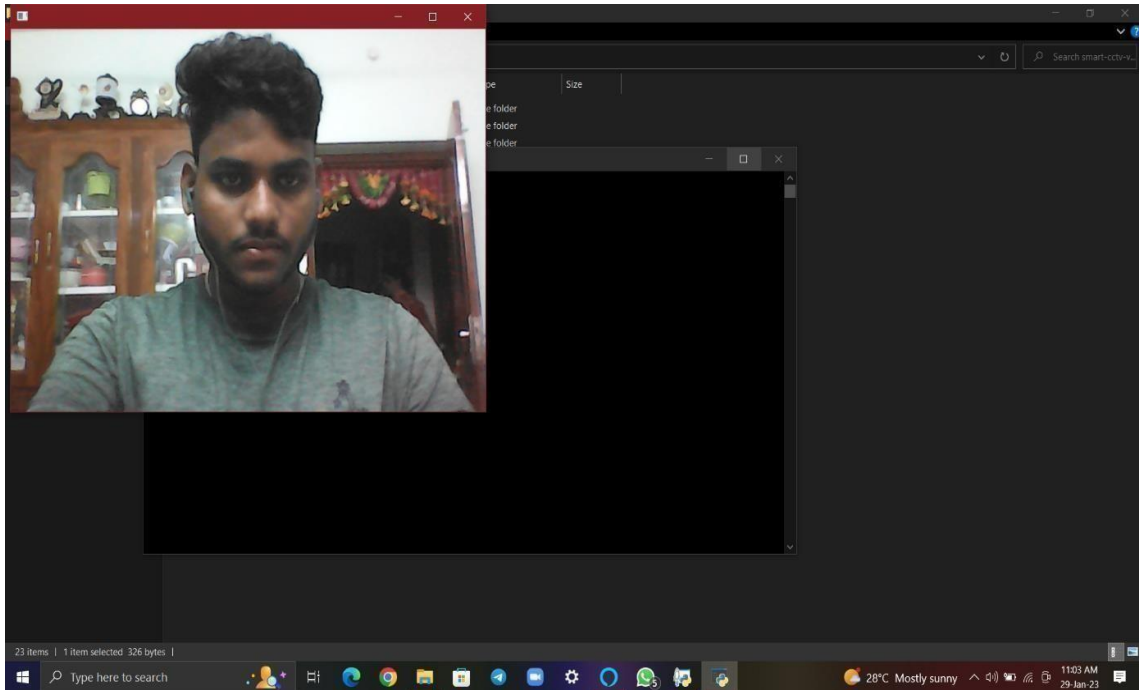


Fig 7.8 Recording using date and time.

Explanation : -

To perform spot the difference detection, the system can be configured to analyze consecutive video frames from the camera using computer vision algorithms. The system can then compare the current frame with the previous frame to detect any changes or differences in the video feed. When a change or difference is detected, the system can generate an alert or notification, indicating a potential security risk or a potential event of interest. Spot the difference detection can be useful in a variety of surveillance applications where subtle changes in the surveillance zone need to be monitored. For example, in a bank or a casino, spot the difference detection can be used to monitor for suspicious behavior, such as someone removing or adding objects from a table or a cash register.

OpenCV is a powerful tool for processing video data. It provides a wide range of functions for reading, writing, filtering, transforming, and analyzing video streams. With OpenCV, you can easily capture video from a webcam or other device, apply various filters to the video frames, and perform complex

transformations such as image registration or image stitching. You can also use OpenCV to analyze video data in real-time, detecting and tracking objects, measuring motion, and more.

Object detection and tracking is a common task in computer vision, and OpenCV provides several algorithms for performing this task. One of the most commonly used algorithms is the Haar cascade classifier, which can be trained to recognize specific objects or patterns in a video stream. Another popular algorithm is the HOG (Histogram of Oriented Gradients) descriptor, which can be used for tasks such as pedestrian detection and tracking. With these algorithms, you can detect and track objects in real-time, and use the resulting data for a variety of applications such as surveillance, robotics, and more.

Machine learning is an important part of modern computer vision, and OpenCV provides support for a variety of machine learning algorithms. For example, you can use OpenCV to train support vector machines (SVMs) for image classification, or to train neural networks for object recognition. With these algorithms, you can build powerful computer vision systems that can recognize and categorize objects in images and video data with a high degree of accuracy.

Real-time computer vision applications are becoming increasingly common in a wide range of fields, from automotive to healthcare to entertainment. With OpenCV, you can develop real-time computer vision applications that can process video data in real-time, detecting and tracking objects, measuring motion, and more. Whether you're building a self-driving car, a medical imaging system, or a virtual reality application, OpenCV provides the tools and algorithms you need to create powerful and reliable computer vision systems.

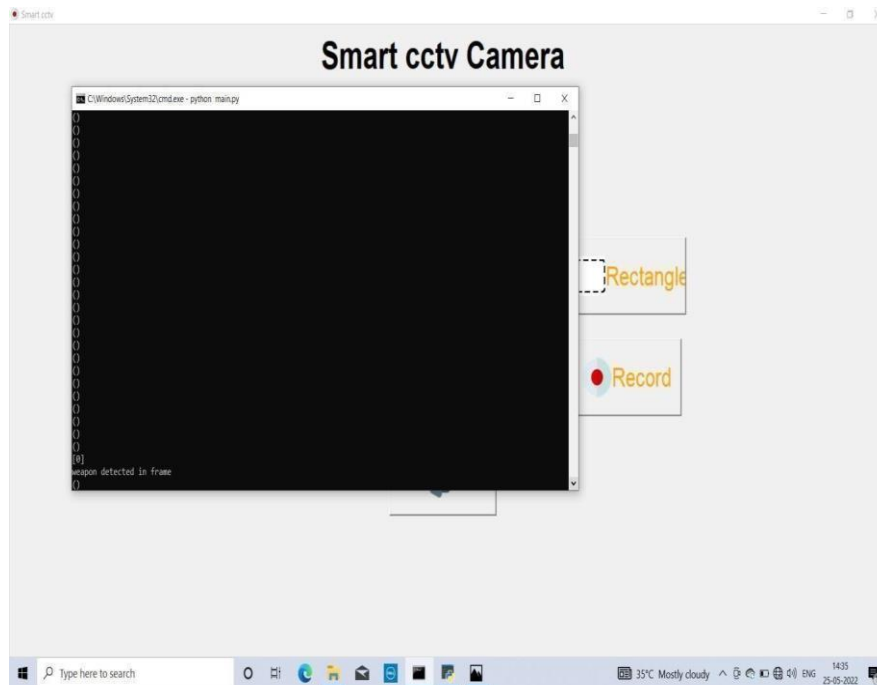


Fig 7.9 Weapon detected text in command prompt

Explanation : -

"Weapon detected" text in command prompt refers to a message displayed on the computer screen or terminal when a video surveillance system using computer vision detects the presence of a weapon in the surveillance zone. This message can be generated by the system when it identifies a specific object or shape that matches the characteristics of a weapon, such as a gun or a knife.

When the "Weapon detected" message is displayed in the command prompt, it indicates that the video surveillance system has detected a potential security threat, and immediate action may be required to respond to the situation. The system may also generate an alert or notification, which can be sent to security personnel or law enforcement agencies to further investigate the situation.

In addition to detecting weapons, a video surveillance system using computer vision can also be programmed to detect other potential security threats, such as suspicious behavior or individuals in the surveillance zone. For example, the system can be configured to detect loitering or trespassing in restricted areas,

or to identify individuals who match a specific description, such as someone wearing a certain type of clothing or carrying a specific object.

To improve the accuracy of weapon detection and other security threat detection, the system can be trained using machine learning algorithms. This involves feeding the system large amounts of training data, which allows the system to learn the characteristics and patterns of weapons and other potential security threats. As the system is trained with more data, its accuracy and reliability in detecting security threats can be improved.

Overall, a video surveillance system using computer vision can be a powerful tool in enhancing security and safety in a variety of settings, from public spaces to high-security facilities. With the ability to detect potential security threats in real-time, the system can help prevent security breaches and respond quickly to potential incidents, ultimately helping to protect individuals and assets. In addition to detecting the presence of a weapon, the system can be programmed to track the movement of the detected object or person to provide additional information about the location and trajectory of the potential threat. This can be particularly useful in scenarios where multiple cameras are used to monitor a large area, as it can help security personnel quickly locate the source of the threat.

To enhance the accuracy of weapon detection, the system can be trained using machine learning algorithms to identify specific patterns and characteristics of weapons. For example, the system can be trained to recognize the shape and size of a gun or a knife, as well as the way these objects are typically held or carried by individuals. It is important to note that weapon detection using computer vision has limitations, and false positives may occur if the system misidentifies an object as a weapon. Therefore, it is essential to configure the system properly and set appropriate thresholds for weapon detection to avoid false alarms and ensure accurate detection.

Weapon detection using computer vision is becoming an increasingly popular approach in the field of security and surveillance, as it can help detect potential threats quickly and accurately. However, it is important to balance the benefits of this technology with privacy concerns, as it may raise questions about the appropriate use of surveillance and the potential impact on civil liberties. Inbuilt function called cascade classifier which we have used in order to detect faces in the frame. The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameter's radius and neighbors. Using the image generated in the last step, we can use the Grid X and Grid Y parameters to divide the image into multiple grids. The model is trained and later on when we want to make predictions the same steps are applied to the image and its histograms are compared with the already trained model.

This feature is used to find the noises in the frames. This is something found in most CCTV but in this module, we'll see how it works. All the frames are continuously analyzed and checked for noises. Noise is checked in the consecutive frames. Simply we do the absolute difference between two frames and in this way the difference between two images is analyzed and Contours (boundaries of the motion are detected). If there are no boundaries then no motion and if there is any there is motion. All images are just integer/ float values of pixels which tell the brightness of pixel and similarly every pixel has values of brightness. So we just do simply absolute difference because negative will make no sense at all. Computer Vision is an interdisciplinary field of study that enables machines to interpret and understand digital images and videos from the world around us. By using techniques such as image processing, pattern recognition, and machine learning, Computer Vision allows machines to acquire, analyze, and understand visual information in a way that is similar to human vision.

CHAPTER – 8

CONCLUSION

In the conclusion of Safety and security are major concerns in the modern era. People employ security mechanisms to safeguard their property be it home or a company. Present security systems involve the use of various sensors for motion detection and cameras for video surveillance i.e., Perimeter Intrusion Detection Systems. This project aims at providing one such idea to ensure safety and security of one's own property. In this paper we propose to provide a Smart CCTV Surveillance system with intrusion detection. Multiple USB cameras are installed at various locations for live streaming and monitoring purposes. This system performs face recognition as an authentication procedure and alerts the owner when an unknown face is detected by sending an email with the snapshot of unknown face and an SMS. Live feed from multiple cameras can be viewed on smartphones and computers. In this project, algorithms are proposed by Sum of Absolute Difference for recording videos only when motion is detected.

The major advantage of this algorithm is that less storage is required as the system records only for frames with a particular threshold. Even the amount of processing required is reduced. The system efficiency is calculated to be around 72%. As recognized images get stored with the name of the person, there is ease of access for surveillance security personal. Thus, the implementation can be done in industrial spaces or places of high security like nuclear power plants, where entry is restricted to authorized persons only. Further, the system can be implemented in farm or forest surveillance where detection or spotting of a specific animal can be recorded and on recognizing potential threat an alarm can be triggered.

REFERENCES

- [1] Anand Sivakumar, Swetha, and R. Gomathi Bhavani. "Image Processing Based System for Intrusion Detection and Home Security Enhancement." In 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 1676-1680. IEEE, 2018
- [2] Bhanse, Vivek Kishor, and M. D. Jaybhaye. "Face Detection and tracking using Image processing on Raspberry Pi." In 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 1099-1103. IEEE, 2018.
- [3] Ch. Husniza Razalli; Mohammed Hazim Alkawaz; Aizat Syazwan Suhemi. Smart IOT Surveillance Multi-Camera Monitoring System. 2019 IEEE 7th Conference on Systems, Process and Control (ICSPC)
- [4] chabadi Narasimha Rao; Reddy Sudheer, Surveillance Camera using IoT and Raspberry
- [5] Gothak Merin Abraham; Nikita Suryawanshi; Nevin Joseph; Dhanashree Hadsul, Future Predicting Intelligent Camera Security System 2021 International Conference on Innovative Trends in Information Technology (ICITIIT)
- [6] Ismail Amir Reza Ebrahimi; Ahmad Reza Naghsh Nilchi; Amir Hassan Monadjemi; Mohammad Saeid Ehsani, IoT based smart surveillance monitoring by using model-based human action recognition
- [7] Javeed Gurusha Lulla; Abhinav Kumar; Govind Pole; Gopal Deshmukh IoT based Smart Security and Surveillance System 2021 International Conference on Emerging Smart Computing and Informatics (ESCI)
- [8] Jahangir Majumder; Joshua Aaron Izaguirre A Smart IoT Security System for Smart- Home Using Motion Detection and Facial
- [9] Wahab Usman Ullahb Electronic Section, UniKL British Malaysian Institute, Malaysia ; Jawad Ali Shah; Kushsairy Kadir; Abdul Wahid Development of Smart Surveillance System using Cloud for Security .
- [10] Wesley Kakadiya, R., Lemos, R., Mangalan, S., Pillai, M. and Nikam, S., 2019, June. Ai based automatic robbery/theft detection using smart surveillance in banks.