### A

**GROUP PROJECT REPORT**

**ON**

**ADVANCED INTELLIGENT VIDEO SURVEILLANCE SYSTEM USING OPENCV**

***Dissertation Submitted In Partial Fulfillment Of The Requirement For The Award Of***

### BACHELOR OF TECHNOLOGY IN

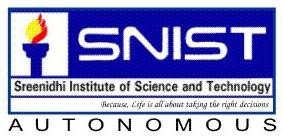
**INFORMATION TECHNOLOGY**

**By**

**D RUTUJA (20311A1208)**

**K KRANTHI REDDY (20311A1210)**

**SISTA MAYUKHA (20311A1244)**



**DEPARTMENT OF INFORMATION TECHNOLOGY SREENIDHI INSTITUTION OF SCIENCE & TECHNOLOGY**

**(An Autonomous Institution) (AFFILIATED TO JNTU, HYDERABAD)**

**Yamnampet, Ghatkesar, R.R. Dist, Hyd-501301.**

**2023**

**ADVANCED INTELLIGENT VIDEO SURVEILLANCE SYSTEM USING OPENCV**

***Dissertation submitted in partial fulfillment of the Requirement for the award of***

### BACHELOR OF TECHNOLOGY IN

**INFORMATION TECHNOLOGY**

**By**

**D RUTUJA (20311A1208)**

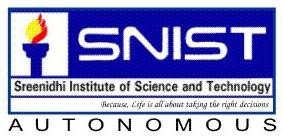
**K KRANTHI REDDY (20311A1210)**

**SISTA MAYUKHA (20311A1244)**

**Under the guidance of**

**Mrs. B Hema Kumari**

Assistant Professor, Dept of IT



**DEPARTMENT OF INFORMATION TECHNOLOGY SREENIDHI INSTITUTION OF SCIENCE AND TECHNOLOGY**

### (An Autonomous Institution) (AFFILIATED TO JNTU, HYDERABAD)

**Yamnampet, Ghatkesar, R.R. Dist, Hyd-501301**

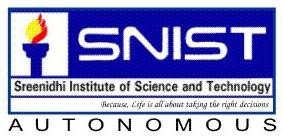
**2022-23**

### DEPARTMENT OF INFORMATION TECHNOLOGY

**SREENIDHI INSTITUTION OF SCIENCE AND TECHNOLOGY**

### (An Autonomous Institution)

**(Affiliated to JNT University; ISO Certified 9001:2000) Yamnampet, Ghatkesar, Hyderabad-501301**

**Ph.Nos:08415-200597, 08415-325444, 9395533303**

### 

### CERTIFICATE

This is to certify that the Dissertation entitled “**ADVANCED INTELLIGENT VIDEO SURVEILLANCE SYSTEM USING OPENCV**”is bonafide workdone and submitted by **D RUTUJA (20311A1208) , K KRANTHI REDDY (20311A1210), SISTA MAYUKHA (20311A1244),** in partial fulfillment of the requirement for the award of Degree of **Bachelor of Technology in Information Technology, SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY, Affiliated to Jawaharlal Nehru Technological University, Hyderabad** is a record of bonafide work carried out by us under the guidance and supervision from **May 2023** to **Sep**  **2023.**

The results presented in this dissertation have been verified and are found to be satisfactory. The results embodied in this dissertation have not been submitted to any other university for the award of any other degree or diploma**.**

**Project Internal Guide Head of the Department Mrs. B Hema Kumari Dr. SUNIL BHUTADA**

Assistant Professor, IT Dept, Department of IT, SNIST, Hyderabad SNIST, Hyderabad

.

**Project Co-ordinator Mr. P. SREEDHAR**

Associate Professor & Associate Head Department of IT, SNIST.

### ACKNOWLEDGEMENT

I thank Dr. **SUNIL BHUTADA**, **Head of the Department of IT** and **MRS. B Hema Kumari,** for supporting us as our **Internal Guide** and providing seamless knowledge over the past one year, and also for providing right suggestion at every phase of the development of our project.

I extend my profound gratitude to thank **Mr. P. Sreedhar**, **(Project Cordinator)**, Department of IT, SNIST, for his valuable suggestion and support throughout the course.

I express a whole hearted gratitude to **Dr. CV TOMY, Director,** and **Dr. Shiva Reddy , Principal, Sreenidhi Institute of Science and Technology** for providing us the conducive environment for carrying through our academic schedules and projects with ease.

I convey my heartful thanks to my Parents, friends, Technical and Non-Technical staff of the college for their constant support in the successful completion of the project.

**D RUTUJA (20311A1208)**

**K KRANTHI REDDY (20311A1210)**

**SISTA MAYUKHA (20311A1244)**

# **ABSTRACT**

The increase in the urban population has resulted in an increase in crime. For citizens, Video surveillance has had a significant influence. Closed Circuit Television (CCTV) is the most widely used system, however it is more expensive and uses more power and storage.  security cameras can be watchful for threats, illegal behaviour, or aggressive behaviour, surveillance will also inevitably record everything in their vision. Dome security cameras, for example, are commonly placed in offices and warehouses. Schools, as another example, typically feature security cameras to monitor parking lots and keep the perimeter secure. However, such surveillance inevitably records everyone ingoing and outgoing through the school. Furthermore, facial recognition and license plate recognition security cameras (LPR) have an extensive capability to focus on specific qualities when within their view. Security cameras deter thieves, but they will not have much effect on those that are determined to commit a crime

**ADVANCED INTELLIGENT VIDEO SURVEILLANCE SYSTEM USING OPENCV**

## **Table of Contents**

### 1. INTRODUCTION 1

* 1. INTRODUCTION TO PROJECT 1
  2. ORGANIZATION PROFILE 1
  3. PURPOSE OF THE SYSTEM 6
  4. PROBLEMS IN EXISTING SYSTEM 6
  5. SOLUTION OF THESE PROBLEMS 6

### 2. SYSTEM ANALYSIS 8

* 1. FUNCTIONAL REQUIREMENTS 8
  2. HARDWARE REQUIREMENT SPECIFICATIONS 8
  3. SOFTWARE REQUIREMENT SPECIFICATIONS 8
  4. MODULES USED IN THE PROJECT 9

### 3. FEASIBILITY REPORT 12

* 1. TECHNICAL FEASIBILITY 12
  2. ECONOMICAL FEASIBILTY 14

### 4. SELECTED SOFTWARE 15

* 1. FEATURES AND DEVELOPMENT 15

### 5. SYSTEM DESIGN 17

* 1. SYSTEM ARCHITECTURE 17
  2. UML DIAGRAMS 18

### 6. CODE 23

### 7. OUTPUT SCREENS 25

**8. SYSTEM TESTING AND IMPLEMENTATION** 26

* 1. TESTING PLAN 26
  2. STRATEGIC APPROACH OF SOFTWARE TESTING 27

### 9. CONCLUSION AND FUTURE SCOPE 29

**10. BIBLIOGRAPHY** 30

**11. APPENDIX C** 31

**12. APPENDIX D** 32

**13. APPENDIX E** 33

**1. INTRODUCTION**

In today’s world, everyone wants their valuables to be safe and secure. The development of the urban population has coincided with an increase in crime. Using video surveillance to monitor a specific region, such as hospitals, institutions, public parks, and buildings, has become necessary. Citizens have been greatly impacted by video monitoring. Video surveillance is required for preventing thefts, monitoring day-to-day activities, protecting property, employee safety, public safety, event video surveillance, traffic monitoring, and so on. Daytime and nighttime video cameras are available in color, monochrome (with or without IR lighting), LLL intensified, thermal IR, analogue and digital, simple and full featured. Cameras with built-in VMD can inform security personnel, improving their ability to detect and locate people as well as be notified to activity on the scene.

All of these types of video cameras continuously monitor and record footage in accordance with the specifications, which consumes more storage space and energy. The cost of keeping these systems up to date is considerably higher. There is no need for continual monitoring in some places where individuals are irregular, like as homes and bank vaults. To address this issue, we developed an advanced intelligent video surveillance system for areas where human presence is irregular. Our system was built using Open CV. It has facial detection models that have been pre-trained.

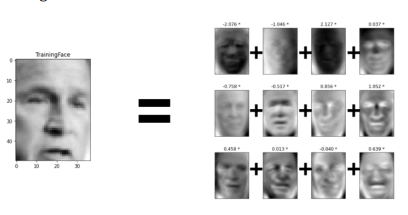
**1.2 ORGANISATION PROFILE**

For detecting the people, we applied Haarcascade classifiers. These classifiers are part of openCV, which we can load into the project by calling a few methods. The proposed system works by capturing video, processing it frame by frame, and then starting recording when it identifies human presence. If the cameras detect any movement, the surveillance system will be activated. The proposed system collects data and saves it locally in a database. The intruder can be identified and apprehended using the video that was collected and saved.

Face Detection Algorithms:

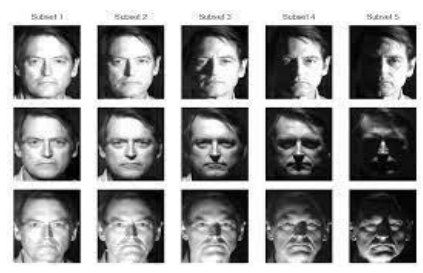
**Eigenfaces:**

In 1991, Turk and Pentland suggested a face recognition technique based on concepts from linear algebra and dimensionality reduction. This method was used in a range of applications at the time, including handwriting recognition, lip reading, medical picture analysis, and others. It is computationally less expensive and straightforward to implement. In 1901, Pearson proposed PCA (Principal Component Analysis) as a method for reducing the number of dimensions. It reduces dimensionality using Eigenvalues and Eigen Vectors.

****

**Fisher faces:**

The eigenfaces have been improved by fisher faces. This approach is predicated on the notion that not every feature of the face is equally important or useful for facial recognition. In order to identify someone when we look at their face, we search for the features that differ the most. The Fisher faces technique has been shown to be 93 percent accurate when used in conjunction with the PCA approach during the pre-processing stage. The fisher faces set of rules has been reported to be 93% accurate when used with the PCA technique on the preprocessing step.

****

**LBPH Algorithm:**

A technique for facial identification called LBPH (Local Binary Patterns Histograms) makes use of LBP. By thresholding each pixel's neighborhood and treating the result as a binary integer, the Local Binary Pattern (LBP) texturing operator assigns labels to individual pixels inside an image. Additionally, it was found that combining LBP with the histograms of oriented gradients (HOG) descriptor considerably improves detection performance on specific datasets. The facial images can be represented by a straightforward data vector. LBP may be used to identify faces because it is a visual descriptor.

Pros

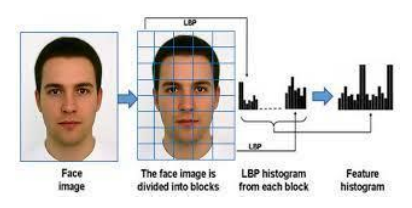
1. Increased precision

2. Easy to understand

Cons

1. Nevertheless, there might be a loss of recognition.

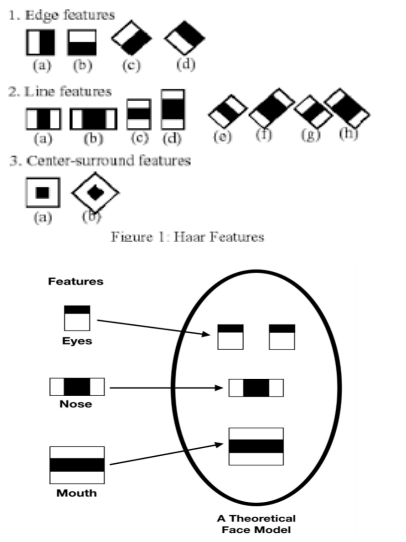
2. It takes a long time to receive acknowledgement.

****

**KERNEL METHODS: PCA AND SVM:**

**PCA:**

PCA (principal component analysis), when applied in the face recognition process, tries to reduce the amount of source data while maintaining the most crucial information. It generates a number of weighted eigenvectors, which in turn generate eigenfaces, vast collections of different images of human faces. Each image in the training set is represented by a linear combination of eigenfaces. These eigenvectors are derived using PCA from the covariance matrix of a series of training images. The major components of each image are calculated (from 5 to 200). The remaining components reveal subtle distinctions between noise and faces. The principal component of the unknown image is contrasted with the major components of all other images as part of the recognition process.



**SVM**

Support vector machine (SVM) is a machine learning technique that uses a two-group classification notion to distinguish between "faces" and "not-faces." To classify fresh test data, a model employs a labelled training data set for each category. Researchers employ both linear and nonlinear. SVM training models for face recognition. According to current results, the nonlinear training machine has a bigger margin and better identification and classification results.

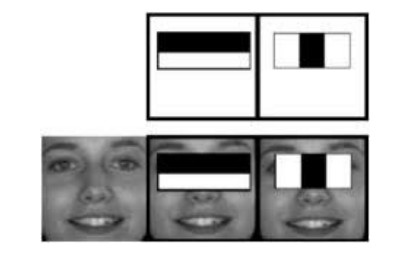
****

Fig:Detecting Facial Features Using Haar

**1.3 PURPOSE OF THE SYSTEM**

The goal of the system was to develop and build a low-cost smart security camera with night vision capabilities. It has person detection capabilities, thus it can prevent crimes. Haarcascader manages the moving body and communicates the collected images. The system makes use of a standard webcam. The data of is stored only when there is a movement of people in the zone where the camera is being fixed. Thus it reduces the storage space by saving only the data when there is movement and detection of human beings.

**1.4 PROBLEMS IN EXISTING SYSTEM**

The existing surveillance system continuously monitors the region and records all the footage.

Drawbacks of Existing System

* The Existing system continuously monitors the area.
* It takes more storage for all the recorded footage
* It uses more electricity
* Maintenance costs are also prohibitively expensive.

**1.5 SOLUTION OF THESE PROBLEMS**

The proposed system is implemented using OpenCV, which is an open source computer vision library. It only monitors and records when a human is identified, reducing both power usage and storage requirements.

The Process steps of our project are:

1. The camera is initially initialized.

2. The camera begins to watch the surrounding area.

3. If the camera identifies a person, the recording will begin.

4. If the person is not detected for a few seconds, the recording will be stopped.

5. The video recordings are saved to a local drive.

Advantages of Proposed System

* It runs efficiently on large databases. To reduce the amount of time that cameras are used to continuously monitors the region.
* In order to save energy.
* To reduce storage utilization.
* To save money on maintenance.

**2. SYSTEM ANALYSIS**

**Functional Requirements**

* Setting up the camera
* Load Haar cascade classifier
* Capture the video
* Store the data
* Setting up the camera:

Firstly, we will set the camera to capture the video.

* Load Haar cascade classifier:

We need to load the haar cascade classifier to detect humans.

* Capture the video:

If human presence is detected, the camera starts recording the video, or else the recording will be stopped.

* Store the data:

If the video is recorded then we will store the data in the local file.

**Software Requirement Specifications**

* Programming Language / Platform: Python
* IDE: Jupyter
* Tools: Anaconda
* Operating System: Windows

**Hardware Requirement Specifications**

* Laptop: WebCam
* RAM: 8GB and Higher.

**MODULES USED IN PROJECT**

1. **Gathering Data:**

Data Gathering is the first step of the machine learning life cycle. The goal of this step is to identify and obtain all data-related problems.

In this step, we need to identify the different data sources, as data can be collected from Kaggle such as csv files. It is one of the most important steps of the life cycle. The quantity and quality of the collected data will determine the efficiency of the output. The more will be the data, the more accurate will be the prediction.

This step includes the below tasks:

* Identify various data sources
* Collect data
* Integrate the data obtained from different sources

1. **Load The Classifier:**

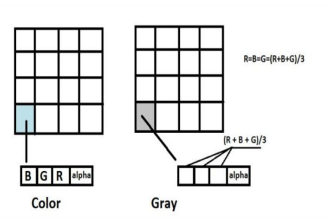
We need to load haar cascade classifier to detect human faces. Face detection will be focused here to train the classifier, the algorithm requires a large number of positive (face related) and negative (non-face related) images. Then we will have to figure out the features we can get out of it. Haar Characteristics like the ones presented below are employed for this.

1. **Gray Scaling Images:**

The process of transforming a picture from another color space to grayscale is known as Gray scaling.

• Reduced dimensionality

• Reduces model complexity



1. **Captures The video:**

If human detects then the camera will start recording the video and save the video into local drive, so that we can view the video for further clarifications or to find the intruder.

**5**. **Stops the recording:**

If human presence is unavailable then camera will be stopped recording, that will save recording time and saves power.

1. **Gathering Data:**

Data Gathering is the first step of the machine learning life cycle. The goal of this step is to identify and obtain all data-related problems.

In this step, we need to identify the different data sources, as data can be collected from Kaggle such as csv files.

It is one of the most important steps of the life cycle. The quantity and quality of the collected data will determine the efficiency of the output.

The more will be the data, the more accurate will be the prediction.

This step includes the below tasks:

* Identify various data sources
* Collect data
* Integrate the data obtained from different sources

1. **Load The Classifier:**

We need to load haar cascade classifier to detect human faces. Face detection will be focused here to train the classifier, the algorithm requires a large number of positive (face related) and negative (non-face related) images. Then we will have to figure out the features we can get out of it. Haar Characteristics like the ones presented below are employed for this.

**8.Gray Scaling Images:**

The process of transforming a picture from another color space to grayscale is known as Gray scaling.

• Reduced dimensionality

• Reduces model complexity



Fig : BGR to Gray Image

**9. Captures The video:**

If human detects then the camera will start recording the video and save the video into local drive, so that we can view the video for further clarifications or to find the intruder.

**10. Stops the recording:**

If human presence is unavailable then camera will be stopped recording, that will save recording time and saves power.

**3. FEASIBILITY REPORT**

**3.1 Technical feasibility**

**Python**

The Python programming language is an Open Source, cross-platform, high level, dynamic, interpreted language. The Python 'philosophy' emphasizes readability, clarity and simplicity, whilst maximizing the power and expressiveness available to the programmer. The ultimate compliment to a Python programmer is not that his code is clever, but that it is elegant. For these reasons Python is an excellent 'first language', while still being a powerful tool in the hands of the seasoned and cynical programmer.

Python is a very flexible language. It is widely used for many different purposes. Typical uses include :

* Web application programming with frameworks like Zope, Django and Turbogears
* System administration tasks via simple scripts
* Desktop applications using GUI toolkits like Tkinter or wxPython (and recently Windows Forms and IronPython)
* Creating windows applications, using the Pywin32 extension for full windows integration and possibly Py2exe to create standalone programs. Scientific research using packages like Scipy and Matplotlib.
* The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
* In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or

Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

* Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

**Python libraries for Machine Learning**

**OpenCV**

OpenCV (Open Source Computer Vision Library) is a free software library for computer vision,image processing and machine learning. It supports a wide range of programming languages, including Python, C++, Java, and others. More than 2500 optimised algorithms are included in the openCV library, which comprises a comprehensive mix of both classic and state-of-the-art computer vision and machine learning techniques. It is compatible with a number of libraries, including numpy, which may be used to execute a wide range of mathematical operations. Using openCV, we may create real-time computer vision applications.It primarily deals with image processing. We can read and write images, process images, analyse videos, and capture and store videos using the OpenCV library. Object detection, object tracking, face identification, licence plate reading, photo editing, enhanced robotic vision, and optical character recognition are just a few of the applications. OpenCV has a user group of about 47 thousand members and an estimated number of downloads of over 14 million.. Along with well-known corporations like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, and Toyota that use the library, several startups like Applied Minds, VideoSurf, and Zeitera use it extensively.

**Visual studio code**

Visual Studio Code, often abbreviated as VS Code, is a free and open-source code editor developed by Microsoft. It has gained immense popularity among developers due to its lightweight yet powerful features and extensibility. VS Code is available on multiple platforms, including Windows, macOS, and Linux.

**3.2 Financial Feasibility**

The costs which are required for the setup of the system involves a device to capture the video i.e video surveillance system. The storage space required for storing the data that was been recorded. Generally this system is very affordable and also provides safety, thus is an advanced version of the already existing system.

1. **SELECTED SOFTWARE**

**Features and Development:**

In an advanced surveillance system using OpenCV, several software components work together to achieve various functionalities. OpenCV itself is a crucial library for computer vision tasks, but it often serves as a foundational piece in a larger software ecosystem. Here are some selected software components commonly used in an advanced surveillance system using OpenCV:

1. OpenCV:

OpenCV (Open Source Computer Vision Library) is the core software component responsible for computer vision tasks such as image and video processing, object detection, and facial recognition. It provides a wide range of pre-built algorithms and functions essential for surveillance applications.

2. IP Camera Software:

IP camera software is used to configure and manage network cameras, allowing the surveillance system to capture live video feeds. Some IP camera software also includes advanced features like motion detection, scheduling, and video analytics.

3. Video Management Software (VMS):

A VMS is used to manage and monitor video streams from multiple cameras. It provides functionalities such as video recording, live viewing, playback, and video analytics. VMS platforms often integrate with OpenCV for various computer vision tasks.

4. Object Detection and Tracking Libraries:

In addition to OpenCV, other libraries and frameworks might be used for more advanced object detection and tracking. For example, YOLO (You Only Look Once) and SSD (Single Shot Multibox Detector) are popular deep learning-based object detection frameworks.

5. Facial Recognition Software:

Facial recognition software is used for identifying and verifying individuals from images or video footage. It can be integrated with OpenCV for facial feature extraction and recognition.

6. Motion Detection Software:

Motion detection software uses computer vision algorithms to detect movement in the surveillance footage. This is useful for triggering alerts or capturing specific events.

7. Cloud Services:

For cloud-based surveillance systems, cloud services are employed for storage, processing, and remote access. Cloud-based solutions can be useful for scalability and ease of management.

8. Analytics and Reporting Software:

Advanced surveillance systems often include analytics and reporting software to provide insights into surveillance data. This can include statistical data, anomaly detection, and trend analysis.

9. Remote Monitoring and Control Software:

Software solutions for remote monitoring and control allow authorized personnel to access and manage the surveillance system from anywhere, providing real-time alerts and control capabilities.

10. Mobile Applications:

Mobile applications can be developed to allow users to access the surveillance system on their smartphones or tablets, providing remote viewing and management on the go.

1. **SYSTEM DESIGN**
   1. **SYSTEM ARCHITECTURE**

Firstly, the camera will be turned on and video streaming will be started. The camera will start capturing the video frame by frame. We need to load haar cascade classifier so that we can detect human faces. Then, we need to convert frames into gray scale images. So that we can identify the human easily because gray scale images are one dimensional. If the person is not detected the camera will be turned off, else the camera starts recording and the footage will be saved into the local database.

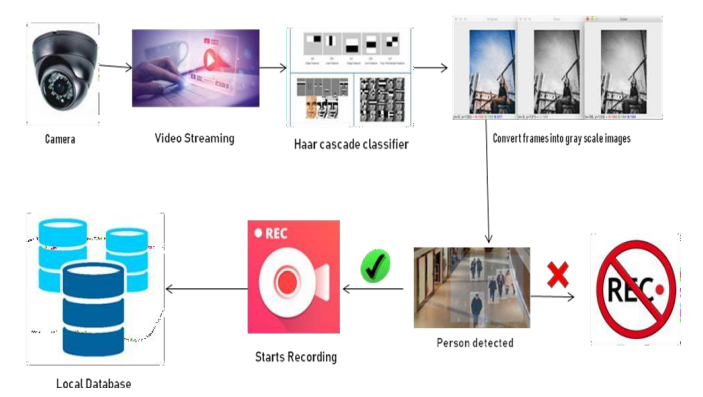


Fig: 5.1 System Architecture

* 1. **UML DIAGRAMS**

1. **USECASE DIAGRAM:**

The purpose of use case diagram is to capture the dynamic aspect of a system. This is used to gather the requirements of a system including internal and external influences. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

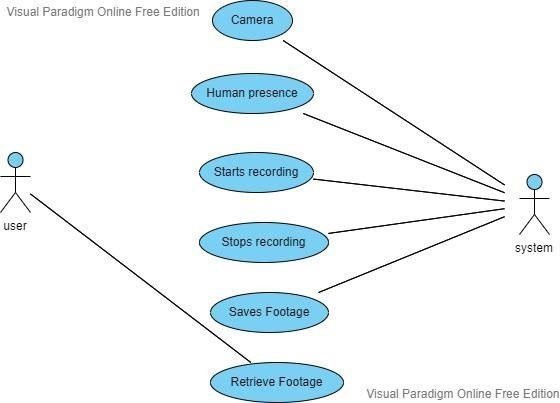


Fig: 5.2 Use Case Diagram

**2 CLASS DIAGRAM:**

The class diagram describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among the classes. It explains which class contains system and also explains the responsibility of the system. This is also known as structural diagram.

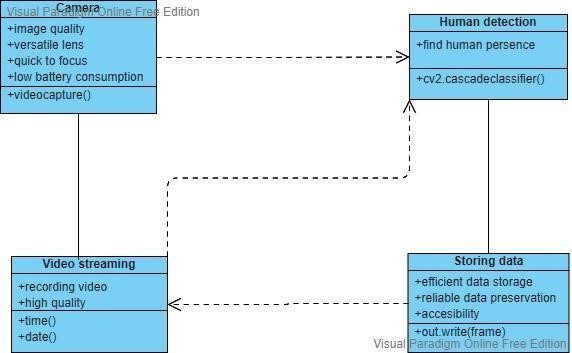


Fig: 5.3 Class Diagram

**3 SEQUENCE DIAGRAM:**

A sequence diagram details the interaction between objects in a sequential order i.e. the order in which these interactions take place.

These diagrams sometimes known as event diagrams or event scenarios. This helps in understanding how the objects and component interacts to execute the process. This has two dimensions which represents time (Vertical) and different objects (Horizontal).



Fig: 5.4 . Sequence Diagram

**4 ACTIVITY DIAGRAM:**

It is behavioural diagram which reveals the behaviour of a system. it sketches the control flow from initiation point to a finish point showing the several decision paths that exist while the activity is being executed.

This doesn’t show any message flow from one activity to another, it is sometimes treated as the flowchart. Despite they look like a flowchart, they are not. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.

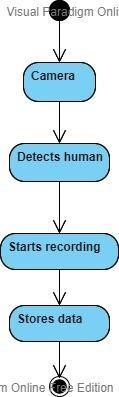


Fig: 5.5 . Activity Diagram

**5 DEPLOYMENT DIAGRAM:**

A deployment diagram in the [Unified Modelling Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language) models the physical deployment of artic rafts on [nodes.](http://en.wikipedia.org/wiki/Node_%28UML%29) To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI). The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers. Device nodes are physically computing resources with processing memory and services to execute software, such as typical computer or mobile phones. An execution environment node (EEN) is a software computing resource that runs within an outer node and which itself provides a service to host and execute other executable software elements.

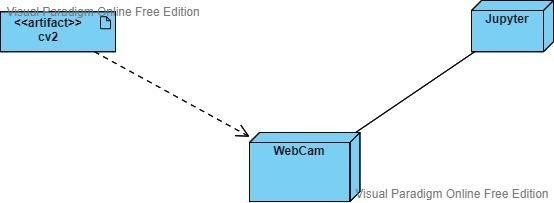


Fig: 5.6. Deployment Diagram

1. **CODE :**

import cv2

import time

import datetime

cap = cv2.VideoCapture(0)

face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +

"haarcascade\_frontalface\_default.xml")

body\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +

"haarcascade\_fullbody.xml")

detection = False

detection\_stopped\_time = None

timer\_started = False

SECONDS\_AFTER\_DETECTION = 5

f\_size = (int(cap.get(3)), int(cap.get(4)))

fourcc = cv2.VideoWriter\_fourcc(\*"mp4v")

while True:

 \_, frame = cap.read() #1--T/F  2---Frame

 gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

 faces = face\_cascade.detectMultiScale(gray, 1.3, 5)

 bodies = body\_cascade.detectMultiScale(gray, 1.3, 5)

 if len(faces) + len(bodies) > 0:

    if detection:

        timer\_started = False

    else:

        detection = True

        current\_time = datetime.datetime.now().strftime("%d-%m-%Y-%H-%M-%S")

        out = cv2.VideoWriter(f"{current\_time}.mp4", fourcc, 20, f\_size)

        print("Started Recording!")

 elif detection:

   if timer\_started:

     if time.time() - detection\_stopped\_time >= SECONDS\_AFTER\_DETECTION:

         detection = False

         timer\_started = False

         out.release()

         print("Recording Stopped!")

   else:

      timer\_started = True

      detection\_stopped\_time = time.time()

 if detection:

   out.write(frame)

 cv2.imshow("Smart Camera", frame)

 if cv2.waitKey(1) == ord('q'):

  break

out.release()

cap.release()

cv2.destroyAllWindows()

1. **OUTPUT:**

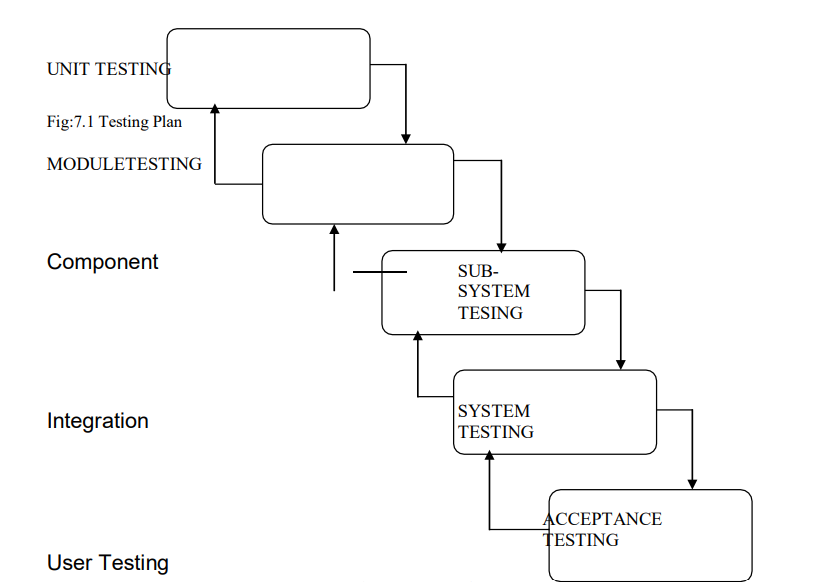




**8. SYSTEM TESTING AND IMPLEMENTATION**

**8.1 TESTING PLAN**

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn. A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code.. Finally, we arrive at system testing, where the software and other system elements are tested as a whole.



**8.2 SOFTWARE TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**TESTING METHODOLOGIES**

The following are the Testing Methodologies:

o Unit Testing. o Integration Testing.

o User Acceptance Testing.

o Output Testing.

o Validation Testing.

o Black box Testing.

o White box Testing

1.Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing. During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

2. Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

1. User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

1. Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

1. White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge ofthe inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

1. Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software work

**9. CONCLUSION AND FUTURE SCOPE:**

The system uses a haar cascade classifier to detect faces. It’s simple to put into action. This approach is particularly beneficial in situations where people are irregular but the region must be monitored. It has the potential to significantly reduce storage use, power consumption, and maintenance costs.

To function properly, the entire system only required the installation of OpenCV. It is a cost effective technology that is simple to implement. This technique may be used by anyone, and it helps to reduce the amount of storage space necessary for the footage. This project is beneficial in any industry where safety is a top priority. Smart Surveillance is important in public spaces, hospitals, and other organizations. This project can be expanded by including encryption techniques to increase surveillance security.

**10. BIBLIOGRAPHY**

1. A. Renjith and Aishwarya, “Enhanced home security using iot and raspberry pi,” International Research Journal of Engineering and Technology (IRJET), vol. 4, 2017.

1. W. F. Abaya, J. Basa, M. Sy, A. C. Abad, and E. P. Dadios, “Low-cost smart security camera with night vision capability using raspberry pi and opencv,” 2014.

1. M. Pervaiz, Y. Y. Ghadi, M. Gochoo, A. Jalal, S. Kamal, and D.-S. Kim, “A smart surveillance system for people counting and tracking using particle flow and modified som,” Sustainability, 2021.

1. M. Rashmika, “Motion sensor and face recognition-based surveillance system using raspberry pi,” International Journal of Advanced Research in Computer Science, vol. 8, no. 5, 2017.

1. A. CobParro, L. Gutierrez, Marron-Romera, Gardel-Vicente, and Bravo-Munoz, “Smart video surveillance system based on edge computing,” 2021.

1. T. Shivprasad, B. Shivani, A. P. Singh, and Deepak, “Survey paper on smart surveillance system,” International Research Journal of Engineering and Technology (IRJET), vol. 3, 2016.

## **APPENDIX C: ABSTRACT**

|  |  |  |
| --- | --- | --- |
| **Sreenidhi Institute of Science and Technology**  **Department of Information Technology**  **Group Project** | | |
| **Batch No: 3** | | **Title** |
| **Roll No** | **Name** |
| 20311A1208 | D Rutuja | **ADVANCED INTELLIGENT VIDEO SURVEILLANCE SYSTEM USING OPENCV** |
| 20311A1210 | K Kranthi Reddy |
| 20311A1244 | Sista Mayukha |

**ABSTRACT**

The increase in the urban population has resulted in an increase in crime. For citizens, Video surveillance has had a significant influence. Closed Circuit Television (CCTV) is the most widely used system, however it is more expensive and uses more power and storage.  security cameras can be watchful for threats, illegal behaviour, or aggressive behaviour, surveillance will also inevitably record everything in their vision. Dome security cameras, for example, are commonly placed in offices and warehouses.

### Project Coordinator Internal Guide Head of the Department

Mr. P. SREEDHAR Mrs. B Hema Kumari Dr. Sunil Bhutada

Associate Professor & Assistant Professor Professor

Associate Head

Department of IT Department of IT Department of IT

|  |
| --- |
| **Student:** D.RUTUJA  K Kranthi Reddy  Sista Mayukha |
|  |

**APPENDIX D: CORRELATION BETWEEN THE GROUP PROJECT AND THE PROGRAM OUTCOMES (POS), PROGRAM SPECIFIC OUTCOMES (PSOS)**

|  |  |  |
| --- | --- | --- |
| **Batch No: 3** | | **Title** |
| **Roll No** | **Name** |
| 20311A1208 | D.RUTUJA | **ADVANCED INTELLIGENT VIDEO SURVEILLANCE SYSTEM USING OPENCV** |
| 20311A1210 | K Kranthi Reddy |
| 20311A1244 | Sista Mayukha |

Table 1: Project/Internship correlation with appropriate POs/PSOs (Please specify level of Correlation, H/M/L against

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **H** | **High** | **M** | **Moderate** | **L** | **Low** |

POs/PSOs)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY DEPARTMENT OF INFORMATION TECHNOLOGY**  **Projects Correlation with POs/PSOs** | | | | | | | | | | | | | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **M** | **L** | **L** | **H** | **H** | **L** | **M** | **H** | **M** | **H** | **H** | **H** | **H** | **H** | **M** |

## 

### Project Coordinator Internal Guide Head of the Department

Mr. P. SREEDHAR Mrs. B Hema Kumari Dr. Sunil Bhutada

Assistant Professor & Assistant Professor Professor

Associate Head

Department of IT Department of IT Department of IT

|  |
| --- |
| **Student:** D Rutuja  K Kranthi Reddy  Sista Mayukha |

## **APPENDIX E: DOMAIN OF PROJECT AND NATURE OF PROJECT**

|  |  |  |
| --- | --- | --- |
| **Batch No: 3** | | **Title** |
| **Roll No** | **Name** |
| 20311A1208 | D.RUTUJA | **ADVANCED INTELLIGENT VIDEO SURVEILLANCE SYSTEM USING OPENCV** |
| 20311A1210 | K Kranthi Reddy |
| 20311A1244 | Sista Mayukha |

Table 2: Nature of the Project/Internship work (Please tick √ Appropriate for your project)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Batch No:** | **Title** | **Nature of Project** | | | |
| **Product** | **Application** | **Research** | **Others**  **(Please Specify)** |
| **3** | ADVANCED INTELLIGENT VIDEO SURVEILLANCE SYSTEM USING OPENCV |  |  |  |  |

### Project Coordinator Internal Guide Head of the Department

Mr. P. SREEDHAR Mrs. B Hema Kumari Dr. Sunil Bhutada

Assistant Professor & Assistant Professor Professor

Associate Head

Department of IT Department of IT Department of IT

|  |
| --- |
| **Student:** D Rutuja  K Kranthi Reddy  Sista Mayukha |

Table 3: Domain of the Project/ Internship work (Please tick √ Appropriate for your project)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Batch No:** | **Title** | **Domain of the Project** | | | | |
| **Artificial Intelligence, Machine Learning and Deep Learning** | **Computer Networks, Information Security, Cyber Security** | **Data Warehousing, Data Mining, Big Data Analytics** | **Cloud Computing, Internet of Things** | **Software Engineering, Image Processing** |
| **3** | ADVANCED INTELLIGENT VIDEO SURVEILLANCE SYSTEM USING OPENCV |  |  |  |  |  |

### Project Coordinator Internal Guide Head of the Department

Mr. P. SREEDHAR Mrs. B Hema Kumari Dr. Sunil Bhutada

Assistant Professor & Assistant Professor Professor

Associate Head

Department of IT Department of IT Department of IT

|  |
| --- |
| **Student:** D Rutuja  K Kranthi Reddy  Sista Mayukha |