

PUBLIC CRIME DETECTION SYSTEM FOR COMMONERS IN INDIA

MINOR PROJECT REPORT

Submitted by

Nishanth.D (RA1911008020042)

Sutharsan.V(RA1911008020016)

BarathBalaji.S(RA1911008020013)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

of

FACULTY OF ENGINEERING AND TECHNOLOGY



SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, RAMAPURAM

GUIDED BY DR.R.RATHNA, A.P

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE
NO.		
	ABSTRACT	3
1.	INTRODUCTION	4
2.	LITERATURE REVIEW	5
3.	SYSTEM ANALYSIS	6
4.	SYSTEM REQUIREMENTS	7
5.	ARCHITECTURE DIAGRAM	12
6.	SYSTEM MODULES	14
7.	IMPLEMENTATION	16
8.	RESULT & DISCUSSION	17
9.	CONCLUSION	18
10.	FUTURE ENHANCEMENTS	19
	APPENDIX 1 (Sample Coding)	20
	APPENDIX 2 (Sample Screen Shot)	23
	REFERENCES	26

ABSTRACT

In present world, Crime is not just limited to stealing things physically. CCTV Footage is crucial evidence against any criminal. It's estimated that India foresees a Crime Index of 44.57 making it the 79th country by Crime Index in 2022. Well planned and structured methods can be used in overcoming these crimes. Our assessment mainly focuses in overcoming these barriers and provide efficient solution to the crimes happening around us. We aim to display proficient strategies in assessing the tools and applications needed for each scenario like house robbery, Bike theft, etc. Finding Solutions and implementation of the work is briefly displayed in this paper. The input is given as image data using real time web cam. Methodologies including the use of OpenCV, and Computer Vision are used in evaluating and finding the robbery

Keywords – Computer vision, tensor flow, OpenCV, pyttsx3, media pipe, OS,date time.

CHAPTER-1

INTRODUCTION

Many surveillance cameras had been mounted in stations, industrial facilities, and town regions to discourage crime. However, because the wide variety of cameras increases, the extent of video records has additionally emerged as massive. This makes it nearly impossible for the police and protection corporations to visually reveal for suspicious behavior and crime through checking and studying pix and motion pictures manually. To deal with this challenge, synthetic intelligence (AI) is getting used within the combat in opposition to crime.

Police offerings round the arena are actively operating to apply AI for video evaluation and crook investigation, with a circulate closer to predictive crime detection and prevention primarily based totally at the idea of proactive, as opposed to reactive, crime fighting. AI learns round one hundred kinds of moves as primary moves in advance: from easy moves which include strolling and stopping, to greater complicated moves and moves which include turning the top to the proper and elevating the left hand.

These primary moves may be identified with a median accuracy of ninety percentage or greater. The proposed Crime Detection System the use of OpenCV and AI includes detecting moves and figuring out the individual's face the use of pc imaginative and prescient. The idea additionally includes fetching the individual's call and different data from the backend database and showing to the consumer for powerful identity with more accuracy. Recognition accuracy of suspicious behavior may be in addition delicate after the device's introduction. This may be accomplished through specifying extra behavioral conditions, which include turning the top left and proper to appearance round, and through converting evaluation parameters, which include every action's duration. The subutility called Robbery detection device alarms whilst an unknown individual enters the digital digicam imaginative and prescient zone. If it detects an unknown individual the device alarms the occupant of the area.

The criminal detection system can be installed in any place. This system will help to identify the criminals accurately and there is also an added advantage in this system. Robbery detection system will help to detect robbery if unknown faces come into the view angle of the camera. The detection system can work in any environment.

CHAPTER 2

LITERATURE SURVEY

“Writing Overview on Criminal Recognizable proof in Mumbai utilizing DBSCAN, Akshay Rathod, Rushikesh Sawant, Ashish Choudhary, Neha Singh”, Crime percentages are expanding consistently in India, with Mumbai being the third among the 19 urban communities for 3 back toback years; protection from wrongdoing should be given expanded need by the public authority aswell as people. In this paper, writing overview of wrongdoing examination utilizing DBSCAN grouping on wrongdoing dataset is finished.

“An Overview on Wrongdoing Event Location and forecast Procedures, Shruti S.Gosavi, Shraddha, S. Kavathekar”, Discoveries of this overview were that when the dataset occasions have more number of missing qualities pre-handling turns into an essential undertaking and wrongdoing doesn't happen consistently across metropolitan scenes however moves in specific regions. In thisway, foreseeing wrongdoing areas of interest is a vital undertaking likewise applying post-handlingwill help in bringing down the pace of violations.

“Writing review on video observation wrongdoing action acknowledgment, K Kishore Kumar, “This paper presents an outline of present and past audits for creating future exploration. The distributed diaries from 2000-2020 were dissected to be aware of the video observation and wrongdoing identification techniques in various areas. A survey of the examined scientists and their strategies are accessible in this paper. This overview is valuable to further develop the wrongdoing location methods utilizing video observation. Besides, it is a helpful device to accumulate data.

“Investigation for wrongdoing counteraction utilizing ICT, Yulihño Ochante-Huamaccto, Francis Robles-Delgado, Michael Cabanillas-Carbonell,” Wrongdoing is a social issue that after therepression of Coronavirus has expanded essentially around the world, which is the reason it is vital to understand what mechanical devices can be utilized to forestall criminal demonstrations. In thecurrent work, a fundamental examination was completed to decide the significance of how to forestall wrongdoing utilizing new data innovations. Fifty examination articles were chosen somewhere in the range of 2015 and 2021. The outcomes got tell explicit data on the best way to forestall wrongdoing utilizing new data innovations.

“Overview on wrongdoing investigation and forecast utilizing information mining methods, H. Benjamin Fredrick David and A. Suruliandi,” Information Mining is the technique which incorporates assessing and analyzing enormous prior data sets to produce new data which might be vital for the association. The extraction of new data is anticipated utilizing the current datasets. Many methodologies for examination and expectation in information mining had been performed. Yet, numerous couples of endeavors has made in the criminal science field. Numerous couples of have taken endeavors for looking at the data this multitude of approaches produce. The police headquarters and other comparative law enforcement offices hold numerous huge data sets of data which can be utilized to anticipate or dissect the criminal developments and crime association in the public. The crooks can likewise be anticipated in view of the wrongdoing information. The fundamental point of this work is to play out a study on the directed learning and unaided learning strategies that has been applied towards criminal ID. This paper presents the review on the Wrongdoing examination and wrongdoing expectation utilizing a few Information Mining procedures.

“Experimental Examination for Wrongdoing Expectation and Anticipating Utilizing AI, Wajiha safat, Sohail asghar, Saira andleeb gillani,” Wrongdoing and infringement are the danger to equity and intended to be controlled. Exact wrongdoing expectation and future anticipating patterns can help to computationally upgrade metropolitan wellbeing. The restricted capacity of people to handle complex data from huge information frustrates the early and precise expectation and anticipating of wrongdoing. The exact assessment of the crime percentage, types and problem areas from past examples makes numerous computational difficulties and open doors. In spite of significant examination endeavors, yet there is a need to have a superior prescient calculation, which direct police watches toward crimes. Past investigations are missing to accomplish wrongdoing gauging and forecast exactness in view of learning models. Consequently, this study applied different AI calculations, specifically, the strategic relapse, support vector machine(SVM), Guileless Bayes, k-closest neighbors (KNN), choice tree, multi-facet perceptron (MLP), arbitrary woodland, and Outrageous Slope Helping (XGBoost), and time series investigation by lengthy transient memory (LSTM) and autoregressive incorporated moving normal (ARIMA) model to fit the wrongdoing information more readily. The exhibition of LSTM for time series examination was sensibly sufficient arranged by greatness of root mean square blunder (RMSE) and mean outright mistake (MAE), on the two informational collections.

“Hypothetical and Observational Examination of Wrongdoing Information, Manisha Mudgal, Deepika Punj and Anuradha Pillai,” Wrongdoing is one of the greatest and overwhelming issues in this day and age and it isn't simply hurtful to the individual in question yet in addition to the local area and government. Because of heightening in wrongdoing recurrence, there is a requirement for a framework that can identify and foresee violations. This paper portrays the rundown of the various strategies and methods used to recognize, investigate and anticipate forthcoming and present violations. This paper shows, how information mining strategies can be utilized to recognize and anticipate wrongdoing utilizing affiliation mining rule, k-implies grouping, choice tree, fake brain organizations and profound learning techniques are additionally made sense of. The majority of the explores are as of now dealing with determining the event of future wrongdoing. There is a requirement for approaches that can deal with constant wrongdoing expectation at high velocity and exactness. In this paper, a model has been recommended that can deal with ongoing wrongdoing expectation by perceiving human activities.

“Wrongdoing expectation and interruption discovery with IoT and AI, Anirudh Kumar Tiwari Prof. (Dr.) Bhavana Narain,” The reason for our work is to plan a model that helps the police in distinguishing wrongdoing areas. We have taken a condition that in the event that any individual is heading off to some place and in the wake of seeing a mishap, when the photograph of that mishap is taken then naturally it will be shipped off closest police headquarters. For this, it is important to have an application planned by us both in the shipper and the recipient. This entire matter will straightforwardly interface the police no sweat the police can arrive at that area. GPS will be utilized for area location. In our work we have gathered dataset with the assistance of computerized camera which is connected with IoT gadget. In initial segment of our paper we have examined the grounds of our work under presentation of wrongdoing, advanced picture handling, GPS and IoT. In second piece of our work we have examined the system of our work here sensor board, GPS setting has been talked about alongside dataset. There is various information assortment advances in the IoT. The most broadly utilized innovation is the Remote sensor organization (WSN) utilizes multi-jumping and self-association to keep up with command over the correspondence hubs.

“Wrongdoing Identification Procedure Utilizing Information Mining and K-Means, Khushabu A. Bokde, Tiksha P. Kakade, Dnyaneshwari S. Tumsare,” Wrongdoings will some way or another impact associations and establishments when happened oftentimes in a general public. Consequently, it appears to be important to concentrate on reasons, elements and relations between event of various wrongdoings and tracking down the most fitting ways of controlling and stay away from additional violations. The fundamental target of this paper is to group bunched violations in view of event recurrence during various years. Information mining is utilized widely with regards to examination, examination and disclosure of examples for event of various wrongdoings. We applied a hypothetical model in view of information mining strategies, for example, bunching and grouping to genuine wrongdoing dataset kept by police in Britain and Grains inside 1990 to 2011. The Hereditary Calculation (GA) is utilized for advancing of Anomaly Recognition administrator boundaries utilizing RapidMiner apparatus.

“Criminal face recognition framework, shiva tamrkar,” Criminal Face Identification project expects to construct a robotized Criminal Face Location framework by turning the human capacity to review minute facial subtleties. Distinguishing proof of lawbreakers at the location of a crime can be accomplished in numerous ways like fingerprinting, Potential observers to a wrongdoing have seen the crook however by and large it may not be imaginable to see the substance of the culprit totally. The Lawbreaker Face Discovery Framework will be worked of a current criminal data set. Information would be given as sketch or a picture and matched against the current data set and results would be given. Criminal record by and large contains individual data about specific individual alongside photo.

“A Criminal Community Detection System Based on Evolving Social Graphs, Xiaoli Wang, Meihong Wang & Jianshan Han”, This paper presents an intelligent criminal community detection system, called ACCDS, to support various criminal event detection tasks such as drug abuse behavior discovery and illegal pyramid selling organization detection, based on evolving social graphs. The system contains four main components: data collection, community social graph construction, criminal community detection and data visualization. First, the system collects a large amount of e-government data from several real communities. The raw data consist of demographic data, social relations, house visiting records, and sampled criminal records.

“Real-time face detection and tracking on mobile phones for criminal detection, Lamiaa A. Elrefaei, Alaa Alharthi”, In this paper a criminal detection framework that could help policemen to recognize the face of a criminal or a suspect is proposed. The framework is a client-server video based face recognition surveillance in the real-time. The framework applies face detection and tracking using Android mobile devices at the client side and video based face recognition at the server side. This paper focuses on the development of the client side of the proposed framework, face detection and tracking using Android mobile devices. For the face detection stage, robust Viola-Jones algorithm that is not affected by illuminations is used. The face tracking stage is based on Optical Flow algorithm. Optical Flow is implemented in the proposed framework with two feature extraction methods, Fast Corner Features, and Regular Features. The proposed face detection and tracking is implemented using Android studio and OpenCV library, and tested using Sony Xperia Z2 Android 5.1 Lollipop Smartphone.

“Performance Evaluation of Cyber Criminal Detection Techniques, K. Veena , K. Meena”, Computer technology has advanced to a greater extent which leads to increase in cyber crime committed in recent years. The detection of cyber crime is not an easy task. From the literature, many researchers used various technologies to detect the cyber crime. In this paper, performance evaluation of various techniques are analysed to determine the cyber criminal. Firstly the detection of synthetic identity theft is checked. Secondly, the intrusion detection is checked using the honey pot security mechanism. Thirdly, the detection is further strengthened using the lie detection technique where the false speech of a person is determined. Finally by analysing the user profile, the detection of cyber crime is done using the clustering techniques. Synthetic Identity Theft method performs better than the remaining methods when considered for evaluation. Experimental results show that comparison of the final list of criminal users and the list of criminals determined, the number of genuine users eliminated are 41 out of 100 users, where as the number of genuine users eliminated from other methods are 16, 36 and 38 only.

“Criminal Face Detection System, Anmariya Babu,Ashna Kollamvalappil Thomas”, In this paper a criminal detection framework that could help policemen to recognize the face of a criminal or a suspect is proposed. The framework is a client side video based face recognition surveillance in the real-time. The framework applies face detection and tracking using pre recorded images model at the client side and video based face recognition at the server side. For the face detection

stage, a robust Viola-Jones algorithm that is not affected by illumination is used. The face tracking stage is based on the Optical Flow algorithm. Optical Flow is implemented in the proposed framework with two feature extraction methods, Fast Corner Features, and Regular Features. The proposed face detection and tracking is implemented using OpenCV library. Experiments show that face tracking using Optical Flow with Regular Features achieves a higher level of accuracy and efficiency than Optical Flow with Fast Corner Features. Face detection achieves a real-time performance through Viola-Jones framework where its detection rates are competitive with some of the best methods to date in terms of both performance and running time. Face tracking provides a solution to handle the real-time conditions and video attributes as a temporal continuity attribute but the time consuming of the most tracking algorithms makes tracking problems an open area of research. The recent research seeks to take advantage of the AdaBoost approach to set an initial window of a target object for a tracking method. Then, the tracking method is responsible for tracking the face by distinctive features, but in a fast manner.

“Face Detection and Recognition for Criminal Identification System, Sanika Tanmay Ratnaparkhi, Aamani Tandasi”, The process of identifying and spotting a criminal is slow and difficult. Criminals, these days are getting smarter by not leaving any form of biological evidence or fingerprint impressions on the crime scene. A quick and easy solution is using state-of-the-art face identification systems. With the advancement in security technology, CCTV cameras are being installed at most of the buildings and traffic lights for surveillance purposes. The video footage from the camera can be used to identify suspects, criminals, runaways, missing persons etc. This paper explores a way to develop a criminal identification system using ML and deep neural networks. The following method can be used as an elegant way to make law enforcement hassle-free.

“Face Sketch-Image Recognition for Criminal Detection Using a GAN Architecture, Sunil Karamchandani, Ganesh Shukla”, One of the important cues in solving crimes and apprehending criminals is matching sketches with digital face images. The problem of matching a forensic sketch to a gallery of mugshot images is addressed in this paper. Feature-based technique is implemented and compared against proposed generative adversarial networks (GANs). The designed GAN projects alternate accuracy for the generator and the discriminator for various batch sizes, dropouts, and learning rate and is able to identify the corresponding image in the CUHK database. Simulation results show that while the feature-based matching fails the holistic method having been trained as an innovation process produced promising results.

“PyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis, Theodoros Giannakopoulos”, Audio information plays a rather important role in the increasing digital content that is available today, resulting in a need for methodologies that automatically analyze such content: audio event recognition for home automations and surveillance systems, speech recognition, music information retrieval, multimodal analysis (e.g. audio-visual analysis of online videos for content-based recommendation), etc. This paper presents pyAudioAnalysis, an open-source Python library that provides a wide range of audio analysis procedures including: feature extraction, classification of audio signals, supervised and unsupervised segmentation and content visualization. pyAudioAnalysis is licensed under the Apache License and is available at GitHub (<https://github.com/tyiannak/pyAudioAnalysis/>). Here we present the theoretical background behind the wide range of the implemented methodologies, along with evaluation metrics for some of the methods.

“Programming Real-Time Sound in Python, Yuri De Pra”, For its versatility, Python has become one of the most popular programming languages. In spite of its possibility to straightforwardly link native code with powerful libraries for scientific computing, the use of Python for real-time sound applications development is often neglected in favor of alternative programming languages, which are tailored to the digital music domain. This article introduces Python as a real-time software programming tool to interested readers, including Python developers who are new to the real time or, conversely, sound programmers who have not yet taken this language into consideration. Cython and Numba are proposed as libraries supporting agile development of efficient software running at machine level. Moreover, it is shown that refactoring few critical parts of the program under these libraries can dramatically improve the performances of a sound algorithm.

“Evaluation of Haar Cascade Classifiers for Face Detection, Rafael Padilla”, In the past years a lot of effort has been made in the field of face detection. The human face contains important features that can be used by vision-based automated systems to identify and recognize individuals. Face location, the primary step of the vision-based automated systems, finds the face area in the input image. An accurate location of the face is still a challenging task. Viola-Jones framework has been widely used by researchers to detect the location of faces and objects in each image. Face detection classifiers are shared by public communities, such as OpenCV. An evaluation of these classifiers will help researchers to choose the best classifier for their particular need. This work focuses of the evaluation of face detection classifiers minding facial landmarks. It is a challenge for vision-based automated systems. It has been an active research area involving several disciplines such as image processing, neural networks, statistics, pattern recognition, anthropometry and computer vision. Vision-based automated systems can apply facial recognition and facial identification in numerous commercial applications, such as biometric authentication, human-computer interaction, surveillance, games and multimedia entertainment.

“Face Recognition Using Haar Cascade Classifier, Varun”, With development of machine learning technology many applications have been revolutionized which earlier used to utilize high amount of resources .Face recognition is a crucial security application .Though this paper we present this application using optimized amount of resources and high efficiency. The face is crucial for human identity. It is the feature which best distinguishes a person. Face recognition is an interesting and challenging problem, and impacts important applications in many areas such as identification for law enforcement, authentication for banking and security system access, and personal identification among others. Face recognition is an easy task for humans but its entirely different task for a computer.

SUMMARY

Above Writings centers around Information Mining, K-Means, IOT, DBSCAN. The sole focal point of Wrongdoing location framework is by utilizing the extraordinary mix of OpenCV and guzzling Man-made brainpower to carry out a Face identification philosophy with ADABOOST calculation and organizing a layered design. This analysis gives that ADABOOST algorithm will provide a immersive analysis in terms of speed and storage complexity for live camera face detection.

PROBLEM STATEMENT

The aim of this project is to help citizens to be safe and secure as in some cases if the criminals are released from the prison and this project can help to monitor their activity as the department can't monitor each and every criminal. As per the statistics the robbery rate has increased where old, aged people are living. The robbery detection system can alert the whole neighborhood so it's a easier way to reduce the robbery rate. This project also can be a major part in women security too. Overall this project can keep the nation safe and secure.

CHAPTER 3

SYSTEM ANALYSIS

It is evident from the research of the existing systems that there is a need for advanced and most feasible version for the crime detection system. The solutions which are currently available may provide a wide area of knowledge and satisfy the urgent needs of people. It's to be believed that the current technologies have multiple tech solutions enabled which can be challenging. Python has been proven to implement latest technologies with the help of modules such as OpenCV for crime detection. Python's easy to understand code complexity and inbuilt modules makes it a feasible choice for programmers. Imbibing the likes of HTML, CSS & BOOTSTRAP, provides users a platform for accessing our application using website.

3.1 Advantages

- Enabling the project with Python improves Productivity and increases Versatility
- ADABOOST algorithm reduces the time complexity of the process.
- Highly accurate
- Provides high security with alarms

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 Software Requirements

- Operating system: Windows8,10,11
- Web browser: Google chrome/Firefox/Opera
- Language (UI) – HTML, CSS, JS, Bootstrap
- Text editor: Visual studio

4.2 Hardware Requirements

- Processor: Intel core I7 – 7th Gen or higher
- Processor speed: Min 1ghz, recommended 2ghz or more
- Camera – 1.3 MP or higher
- Speaker
- Memory (RAM): Min 8GB, recommended 8GB or above.

CHAPTER 5

ARCHITECTURE DIAGRAM

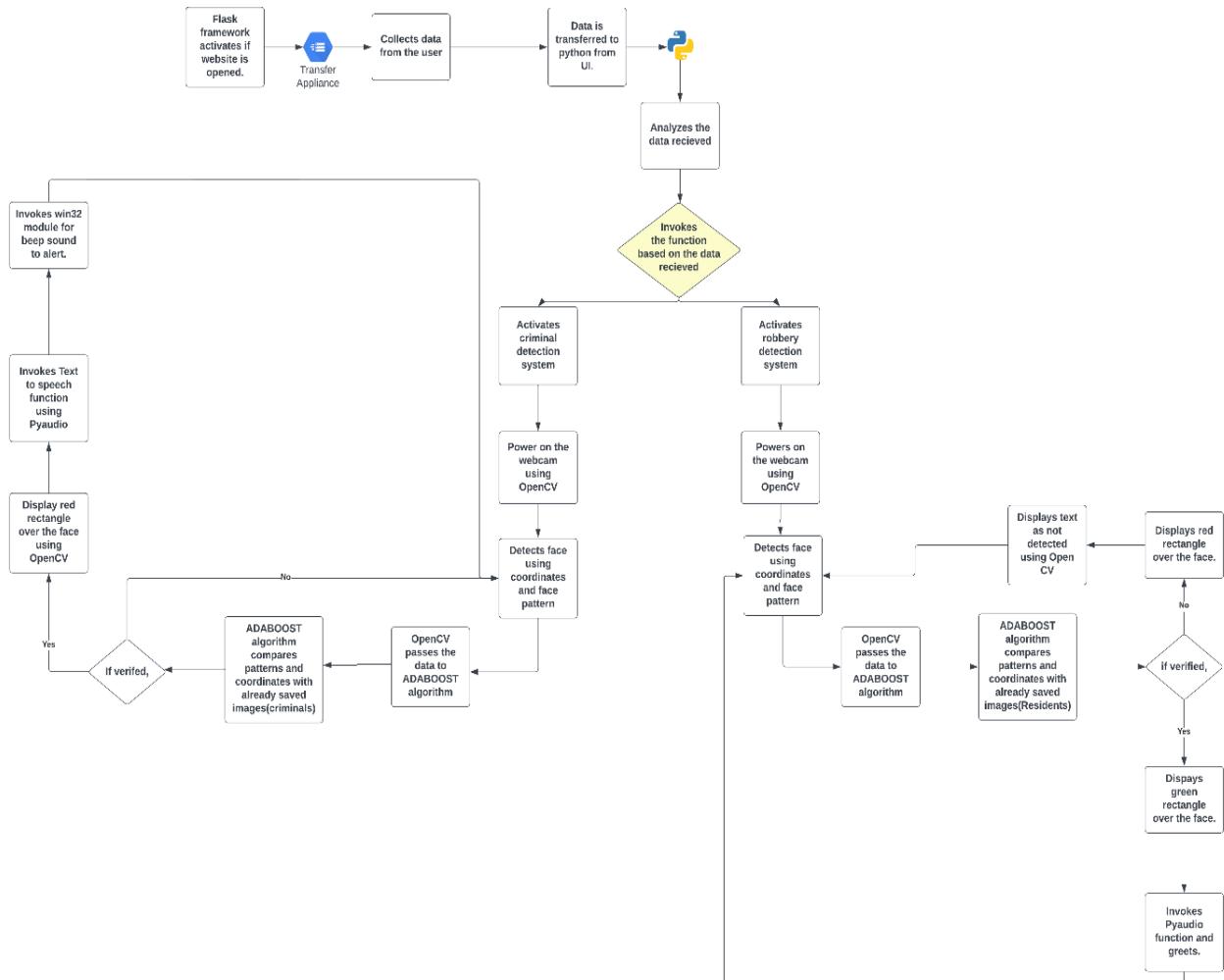


Fig 5.1 Architecture Diagram

Our system's main component is the homepage(website). With reference to Fig. 5.1, The user analyzes the details and can choose a suitable application based on their requirements. Each application has its bifurcations depending upon the domain. Similarities in opening the webcam applies to both Robbery and Criminal Detection system. Criminal Detection system displays the name of the criminal and issues a warning to the user whereas Robbery Detection system voices out a beep sound on detection of robbery. Finally, the application closes when the user clicks "q" in their keyboard.

CHAPTER 6

SYSTEM MODULES

1.Flask module:

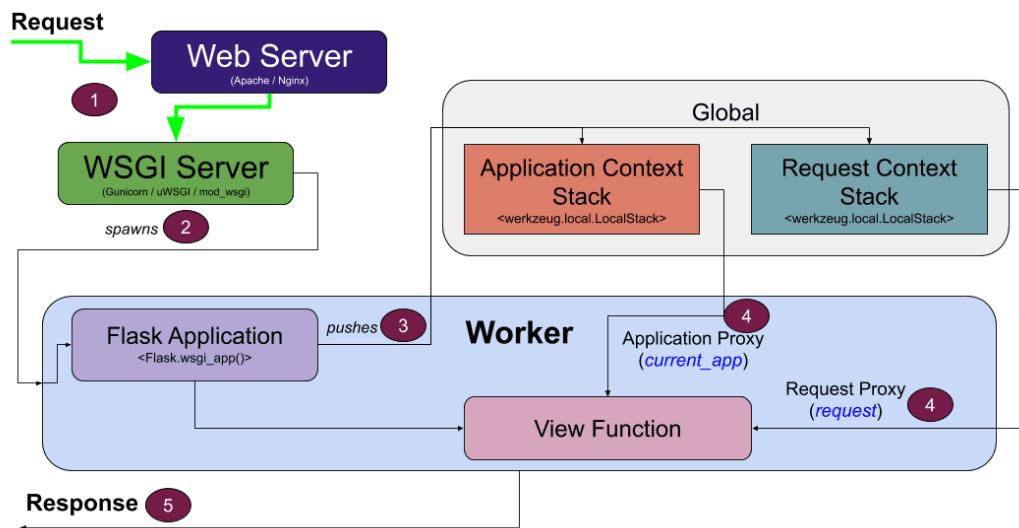


Fig 6.1 Flask module architecture diagram

Flask is a microframework written in python and it's used for web applications which requires minimal operations and high performance. Flask is easy to use. When the user opens the website, The website will send the data to web server and web server sends response to the WSGI server. WSGI server is an important component as it is used to send requests from web server to python(backend). WSGI server is known as Web server gateway interface. The data is received by python application and the data is analyzed respective to the purpose of the application. The further process will be taken care of GET and POST requests simultaneously. GET request is used to collect the input from the user. It also deletes all representation of the target resource specified in the link. The data is collected through GET and transfers to the view function. We can analyze the mentioned process and features in the above diagram Fig 6.1.

2.ADABOOST module:

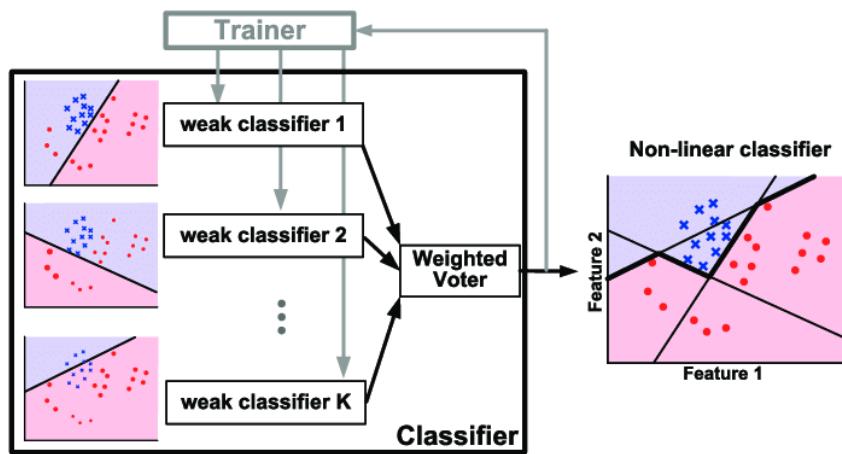


Fig 6.2 ADABOOST Architecture diagram

ADABOOST algorithm is known as adaptive boost algorithm. It is used to boost the python applications. This algorithm achieves high accuracy. This algorithm has a preprogrammed trainer inbuilt with weak classifiers. The weak classifiers help to classify the images in the application. It's a supervised learning and it has already pre trained data inserted. OpenCV has a module named HAAR Cascade classifier. ADABOOST is hidden under HAAR cascade classifier. The HAAR cascade is a powerful algorithm which uses edge or line detection to detect the face. The result is stored in xml files and its read by Open CV methods. ADABOOST is a extension to the HAAR cascade to boost the algorithm. This algorithm is not so complex and can run in reality. We can train a haar-cascade detector to detect various objects and things. The HAAR features needed a lot of mathematical calculation to traverse the face. Haar Cascade is a machine learning-based approach where a lot of positive and negative images, algorithms are used to train the classifier. We can see that the weak classifiers are connected to weighted voter which gives out a graph of non-linear classifier where there are 2 axis called as feature 1 and feature 2. The features and architecture is mentioned clearly in FIG 6.2.

3. Open CV module:

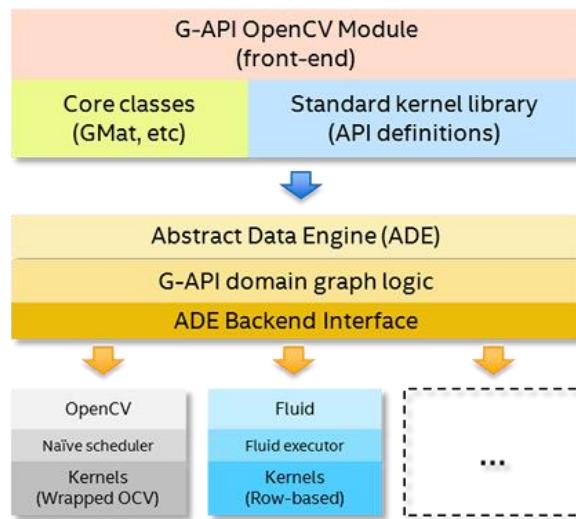


Fig 6.3 Open CV Architecture diagram

Open CV is a python module which is used in applications which uses face comparisons. OpenCV has a front end of G-API OpenCV Module. The front end uses Core classes of GMat and standard kernel library of API definitions. The data is sent to abstract data engine (ADE). The data is than analyzed using G-API domain graph logic and ADE Backend interface. The data than splitted into Naive scheduler and wrapped OCV. The HAAR Classifiers comes into action, and it classifies the data. In simple words, Open CV is a computer vision library with APIs that let you setup a pipeline for your computer vision project. It also provides implementation for many computer vision algorithms along with image algorithms. The process is shown in Fig 6.3 Open CV architecture diagram.

4. Win32API module:

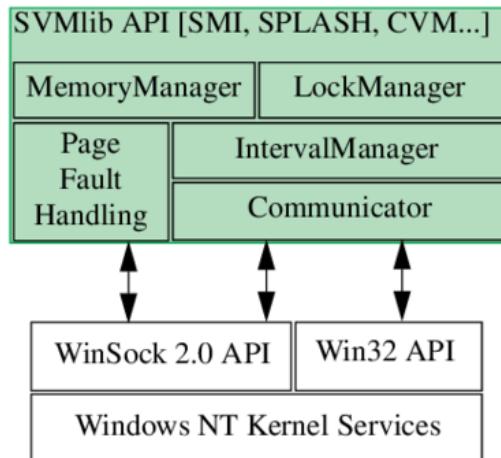


Fig 6.4 Win32API Module

Win32API is a module of python library which gives the services of windows in the program. This is the only API which allows access to windows features. This API is developed by Microsoft and upgrades when windows updates. The main functions of this module are kernel functions, services, create process, etc. We have used this API in this project for activating the beep and buzzer sound. The connection to the features and services of win32api is shown in the diagram fig 6.4 clearly.

CHAPTER 7

IMPLEMENTATION

Create virtual environment and implement flask:

- Create a virtual environment in any of the environment (VS Code, PyCharm, etc.) and install flask(a python web framework) using *pip install flask* command.
- Install all the necessary files like NumPy, pytsxs3, win32api, pythoncom, etc.
- Import the installed files.
- Initialize flask configuration to a variable called as app.
- Initialize classifiers and cascades of OpenCV to a variable

Create robbery detection system:

- Call the video capture function to implement the webcam.
- Implement open cv functions to detect face and mark the coordinates.
- Compare the points and coordinates, the face patterns with the face registered in database using adaboost algorithm.
- Implement the logic to mark the colored rectangle over the face.
- Activate the buzzer to produce beep sound using win32api.
- Display the test as ‘detected’ over the marked area.
- If the face didn’t matched means displays red rectangle with the beep sound.
- If the face matches mean display green rectangle with a audio of welcoming using pyaudio.
- Display all the configurations using imshow () function.
- After the completion of function use destroyAllWindows () to destroy the function cache and data.
- Refer Appendix 1 – figure 1.1 for detailed view.

Create criminal detection system:

- Call the video capture function to implement the webcam.
- Implement open cv functions to detect face and mark the coordinates.
- Compare the points and coordinates, the face patterns with the face registered in database using adaboost algorithm.
- Implement the logic to mark the colored rectangle over the face.
- Activate the buzzer to produce beep sound using win32api.
- If the face didn’t match means no action will takes place.

- If the face matches mean displays red rectangle with a audio of “Criminal detected” following a beep sound.
- Display all the configurations using imshow () function.
- After the completion of function use destroyAllWindows () to destroy the function cache and data.
- Refer Appendix 1 – figure 1.2 for detailed view.

Implementation of flask:

- Route the functions to the respective pages and add parameters of ‘GET’ and ‘POST’.
- Receive the data from the user and analyze the input.
- The input will be as of number
- Invoke the system respect to the input the user has entered.
- Refer Appendix 1 – figure 1.3 for detailed view.

Create UI:

- Create a HTML file named as index.
- Create external files for CSS, Bootstrap.
- Link all the style sheets to the html file.
- Link to the python function using flask framework by creating form which links to the python function.
- Add java script files for speed and animations for the website.
- Create folder called images and add the required images needed for the website.
- Refer the screenshots which are in Appendix 2 for detailed insight.

Create landing page:

- The website should be displayed in the URL – <http://127.0.0.1:5000/>.

Create databases:

- Create folder to store images for the two systems.
- The images are used to compare the live feed for the webcam.

Test the system:

- Test the system using various test cases.
- Test the system in various types of intensity of light.
- Test the system in various angles and different kind of faces too.

Deploying the system in PythonAnywhere:

- Create a git hub repository.
- Import the system in GitHub.
- Connect the repository to python anywhere account.
- Enter some commands to deploy the files from GitHub to python anywhere.

CHAPTER 8

RESULT & DISCUSSION

Ada boost Classifier algorithm combines weak classifier algorithms to create a strong classifier. The accuracy rate of this algorithm will be high when comparing to other algorithms. After training a classifier at any level the Adaboost algorithm assigns weight to each classifier based on accuracy to categorize the classifier into weak and strong.

Formula for Ada boost classifier:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

$H_t(x)$ is the output of weak classifier t for input x . α_t is weight assigned to classifier.

$\alpha_t = 0.5 * \log((1-E)/E)$ – This formula is based on the error rate E .

The accuracy can be calculated as –

$(\text{Total_positive} + \text{Total_negative}) / (\text{Total})$

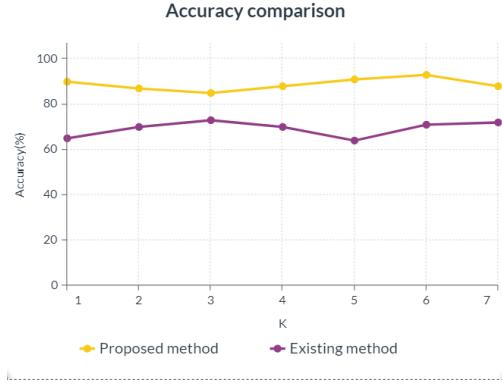


Fig 8.1- Accuracy comparison

Fig 8.1 shows a clear comparison of the accuracies between the proposed system and the existing system. This graph also depicts the accuracy percentages of both the systems. The Adaboost algorithm efficiency is improvised as we can depict from the above graph. The accuracy of the proposed system starts at 90% and we can see the rise from level 3 to level 6. The majority value is 95%. The existing system starts at 63% and we can depict a downfall from level 3 to 5. The majority value is 75% for the existing system. This proof depicts that Adaboost algorithm provides high performance in accuracy and efficiency.

TABLE 1 - Performance metrics

Learning rate	Test accuracy (%)
0.1	95
0.2	75
0.3	94
0.4	98
0.5	98
0.6	94
0.7	99
0.8	97
0.9	93
1	83

The table 1 gives the values for the performance of the system in respect to learning rate. As per the table we can easily depict that the accuracy of the algorithm used in the system. The high accuracy is at learning rate of 0.4 and 0.5.



Fig 8.2-Performance metrics

As we can see the performance metrics are in the range of 80%-100% accuracy. The performance metrics are high in percentage.

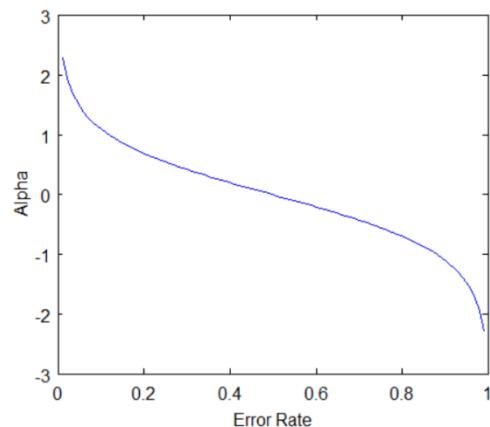


Fig 8.3 – Error rate of proposed system

The error rate is decreasing as alpha decreases. The proposed system is a good choice for crime detection and web cam applications. Hereby, I conclude that the proposed system is accurate and approximate.

CHAPTER 9

CONCLUSION

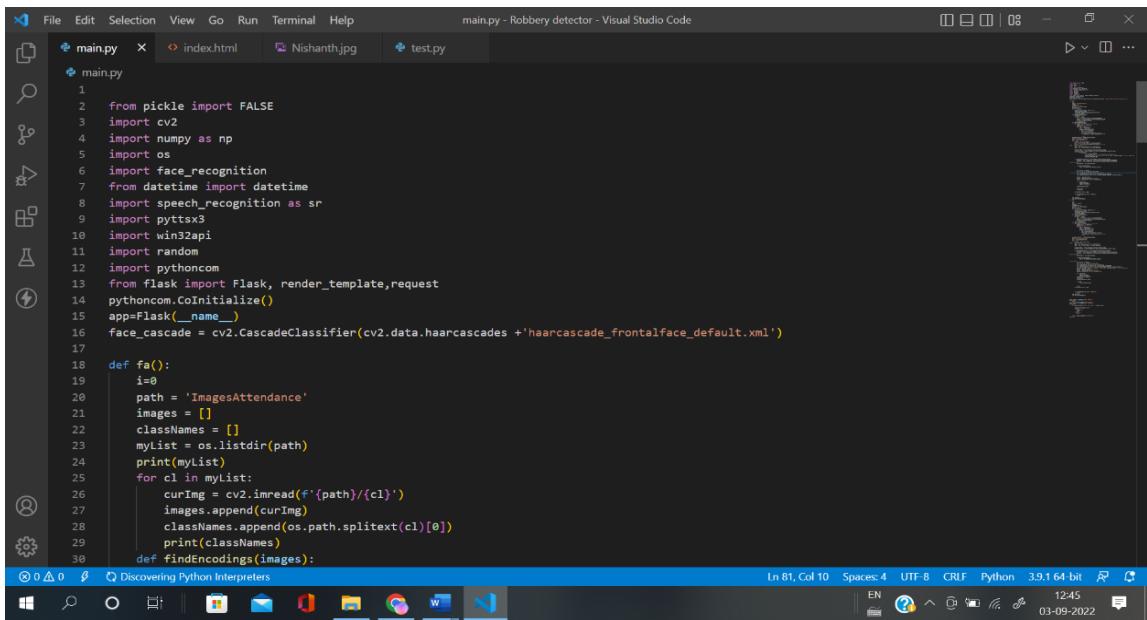
The public crime detection system for commoners helps all kind of public in residential areas and in public places too. This system is highly effective and gears the nation security above the existing systems. This system is also easy to maintain, and it gives an ease of access to all the users. This paper explains about the deployment of criminal detection system, Modules integrations, Webcam integrations and much more. This is a very useful project as it can protect our homes and nation. The website is quick responsive, fast, and feasible.

CHAPTER 10

FUTURE ENHANCEMENTS

- The addition of systems like bike theft detection system, Pickpocket detection system, etc. will be the enhancements for this project.
- Advanced machine learning algorithms will be implemented to reinforce accurate predictions.
- The system will be reinforcing with certain advanced optimization techniques to improve speed and reduce page loading time.
- The ease of access will be improved.

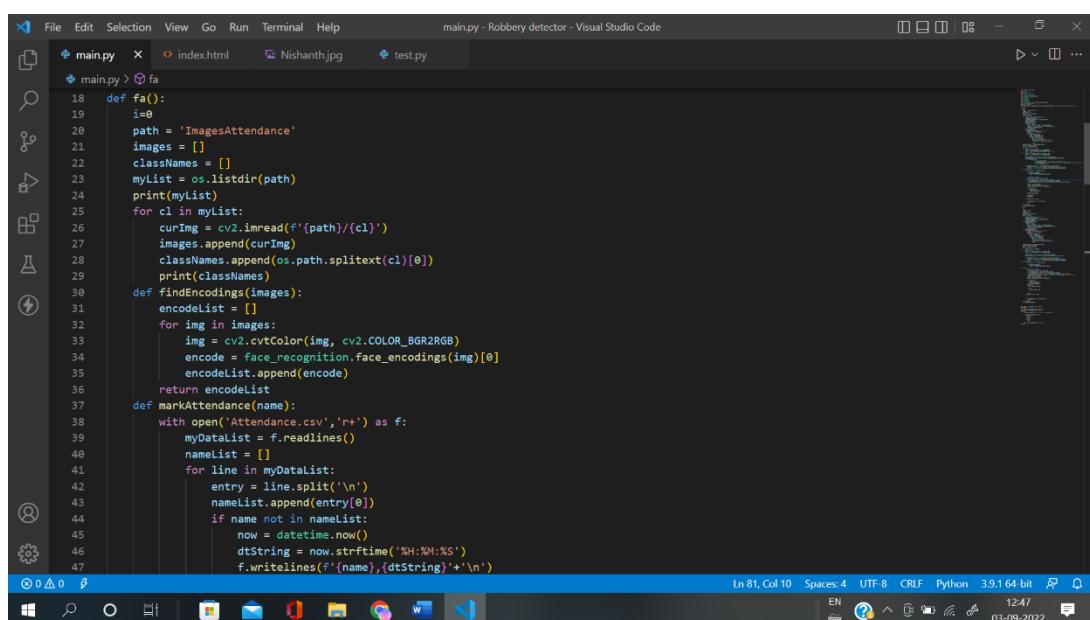
APPENDIX 1 (SAMPLE CODING)



```
File Edit Selection View Go Run Terminal Help main.py - Robbery detector - Visual Studio Code
main.py index.html Nishanth.jpg test.py
main.py
1
2  from pickle import FALSE
3  import cv2
4  import numpy as np
5  import os
6  import face_recognition
7  from datetime import datetime
8  import speech_recognition as sr
9  import pyttsx3
10 import win32api
11 import random
12 import pythoncom
13 from flask import Flask, render_template, request
14 pythoncom.CoInitialize()
15 app=Flask(__name__)
16 face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
17
18 def fa():
19     i=0
20     path = 'ImagesAttendance'
21     images = []
22     classNames = []
23     myList = os.listdir(path)
24     print(myList)
25     for cl in myList:
26         curImg = cv2.imread(f'{path}/{cl}')
27         images.append(curImg)
28         classNames.append(os.path.splitext(cl)[0])
29         print(classNames)
30     def findEncodings(images):
31         encodeList = []
32         for img in images:
33             img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
34             encode = face_recognition.face_encodings(img)[0]
35             encodeList.append(encode)
36         return encodeList
37     def markAttendance(name):
38         with open('Attendance.csv', 'r+') as f:
39             myDataList = f.readlines()
40             nameList = []
41             for line in myDataList:
42                 entry = line.split(',')
43                 nameList.append(entry[0])
44             if name not in nameList:
45                 now = datetime.now()
46                 dtString = now.strftime('%H:%M:%S')
47                 f.writelines(f'\n{name},{dtString}')


In 81, Col 10  Spaces: 4  UTF-8  CRLF  Python  3.9.1 64 bit  12:45  03-09-2022
```

Fig 1.1 (A) – Importing necessary packages



```
File Edit Selection View Go Run Terminal Help main.py - Robbery detector - Visual Studio Code
main.py index.html Nishanth.jpg test.py
main.py > fa
18 def fa():
19     i=0
20     path = 'ImagesAttendance'
21     images = []
22     classNames = []
23     myList = os.listdir(path)
24     print(myList)
25     for cl in myList:
26         curImg = cv2.imread(f'{path}/{cl}')
27         images.append(curImg)
28         classNames.append(os.path.splitext(cl)[0])
29         print(classNames)
30     def findEncodings(images):
31         encodeList = []
32         for img in images:
33             img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
34             encode = face_recognition.face_encodings(img)[0]
35             encodeList.append(encode)
36         return encodeList
37     def markAttendance(name):
38         with open('Attendance.csv', 'r+') as f:
39             myDataList = f.readlines()
40             nameList = []
41             for line in myDataList:
42                 entry = line.split(',')
43                 nameList.append(entry[0])
44             if name not in nameList:
45                 now = datetime.now()
46                 dtString = now.strftime('%H:%M:%S')
47                 f.writelines(f'\n{name},{dtString}')


In 81, Col 10  Spaces: 4  UTF-8  CRLF  Python  3.9.1 64 bit  12:47  03-09-2022
```

Fig 1.1(B) – Robbery detection system

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows files: main.py, index.html, Nishanth.jpg, and test.py.
- Code Editor:** Displays Python code for a criminal detection system. The code includes functions for reading images from a directory, extracting face encodings, and marking attendance in a CSV file.
- Status Bar:** Shows the current file is main.py - Robbery detector - Visual Studio Code, line 111, column 30, spaces: 4, encoding: UTF-8, Python 3.9.1 64 bit, and the date/time: 03-09-2022 12:58.

Fig 1.1(C) – Criminal detection system

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows files: main.py, index.html, Nishanth.jpg, and test.py.
- Code Editor:** Displays Python code using the Flask framework. It defines routes for home and tt, and handles POST requests for tt.
- Status Bar:** Shows the current file is main.py - Robbery detector - Visual Studio Code, line 111, column 30, spaces: 4, encoding: UTF-8, Python 3.9.1 64 bit, and the date/time: 03-09-2022 12:59.

Fig 1.1(D) – Flask framework

The screenshot shows the Visual Studio Code interface with the index.html file open. The code is a template for a web page, starting with a DOCTYPE declaration and an HTML structure. It includes meta tags for character encoding, title, viewport, keywords, and description. It also links to favicon.ico and font-awesome.css files. The body section contains a row with padding and margin settings. The status bar at the bottom shows the file path, line count (Ln 347), column count (Col 8), and other settings.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>CYBERSPY</title>
    <meta content="width=device-width, initial-scale=1.0" name="viewport">
    <meta content="Free Website Template" name="keywords">
    <meta content="Free Website Template" name="description">
    <!-- Favicon -->
    <link href="{{url_for('static',filename='img/favicon.ico')}}" rel="icon">
    <!-- Font Awesome -->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css" rel="stylesheet">
    <!-- Flaticon Font -->
    <link rel="stylesheet" href="{{url_for('static',filename='lib/owlcarousel/assets/owl.carousel.min.css')}}" rel="stylesheet">
    <!-- Libraries Stylesheet -->
    <link rel="stylesheet" href="{{url_for('static',filename='lib/owlcarousel/assets/owl.carousel.min.css')}}" rel="stylesheet">
    <!-- Customized Bootstrap Stylesheet -->
    <link rel="stylesheet" href="{{url_for('static',filename='css/style.css')}}" rel="stylesheet">
</head>
<body class="bg-white">
    <!-- Topbar Start -->
    <div class="row py-3 px-lg-5">
```

Fig 1.2(A) – UI

The screenshot shows the Visual Studio Code interface with the index.html file open. The code includes sections for 'Subscribe Start' and 'Subscribe End'. The 'Subscribe Start' section contains instructions for users to click images to see the app in action. The 'Subscribe End' section starts with a 'Services Start' heading, followed by a card component for 'High tech systems' with a title, subtitle, and a form for a POST request. Below this is another card for 'Criminal detection system' with a button for selecting an image and a footer note about the database. The status bar at the bottom shows the file path, line count (Ln 347), column count (Col 8), and other settings.

```
<!-- Subscribe Start -->
<div class="container-fluid bg-info my-5 py-5 text-center">
    <h1 class="text-white font-weight-bold mt-5 mb-3">Click the below images to see the app in action</h1>
    <p class="text-white mb-4">After clicking there will be a window popping in your start bar...open the window and use it...<img alt="camera icon" style="vertical-align: middle;"></p>
    <p class="text-white mb-4">Kindly note that the start of the application will take time to detect the camera...Be patient to witness th
</div>
<!-- Subscribe End -->
<!-- Services Start -->
<div class="container pt-5">
    <div class="d-flex flex-column text-center mb-5">
        <h5 class="text-primary mb-3">Our Applications</h5>
        <h1 class="m-0">High tech systems</h1>
    </div>
    <form method="POST" action="#">
        <div class="row">
            <div class="col-md-6 col-lg-5 mb-4">
                <div class="card mb-2 p-3">
                    <button name="numb" value="2">
                    <div class="card-body bg-secondary d-flex align-items-center p-0">
                        <h3 class="flaticon-desk font-weight-normal d-flex flex-shrink-0 align-items-center justify-content-center bg-prima
                        <h6 class="card-title text-white text-truncate m-0">Criminal detection system</h6>
                    </div>
                    <div class="card-footer">
                        Criminal detection system is connected to criminals database contains criminal images which recognizes and detects
                        detects face it blares an alarm with a saying "criminal beware".Press <b>Q</b> to exit the application.
                    </div>
                </div>
            </div>
        </div>
    </form>
</div>
```

Fig 1.2(B)- UI

The screenshot shows the Visual Studio Code interface with the 'index.html' file open. The code includes various script tags for jQuery, Bootstrap, and other static files, along with CSS and HTML templates.

```
<!-- Back to Top -->
<a href="#" class="btn btn-secondary border back-to-top"><i class="fa fa-angle-double-up">/<i></a>

<!-- JavaScript Libraries -->
<script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.bundle.min.js"></script>
<script src="{{url_for('static',filename='lib/easing/easing.min.js')}}"/></script>
<script src="{{url_for('static',filename='static/lib/waypoints.waypoints.min.js')}}"/></script>
<script src="{{url_for('static',filename='lib/counterup/counterup.min.js')}}"/></script>
<script src="{{url_for('static',filename='lib/owlcarousel/owl.carousel.min.js')}}"/></script>

<!-- Contact Javascript File -->
<script src="{{url_for('static',filename='mail/jqBootstrapValidation.min.js')}}"/></script>
<script src="{{url_for('static',filename='mail/contact.js')}}"/></script>

<!-- Template Javascript -->
<script src="{{url_for('static',filename='js/main.js')}}"/></script>

</body>
</html>
```

Fig 1.2(C) – HTML file

The screenshot shows the Visual Studio Code interface with the 'style.scss' file open. The code defines SCSS styles for various components like testimonial-carousel and contact-form.

```
.testimonial-carousel .owl-dot {
    display: inline-block;
    margin: 0 5px;
    width: 12px;
    height: 12px;
    border-radius: 10px;
    background: #dddddd;
}

.testimonial-carousel .owl-dot.active {
    background: $primary;
}

.contact-form .help-block ul {
    margin: 0;
    padding: 0;
    list-style-type: none;
}

@media (max-width: 767.98px) {
    .contact-form {
        margin: 30px 0 45px 0;
    }
}
```

Fig 1.2(D) – CSS file

```

static > scss > style.scss
36 h2,
37 h3,
38 h4,
39 h5,
40 h6 {
41   font-weight: 700 !important;
42   text-transform: uppercase;
43 }
44 [class^="flaticon-"]:before,
45 [class^=" flaticon-"]::before,
46 [class^="flaticon-"]::after,
47 [class^=" flaticon-"]::after {
48   font-size: inherit;
49   margin-left: 0;
50 }
51 .back-to-top {
52   position: fixed;
53   display: none;
54   right: 30px;
55   bottom: 30px;
56   z-index: 11;
57   -webkit-animation: action 1s infinite alternate;
58   animation: action 1s infinite alternate;
59 }
60 }
61 }

```

Fig 1.2(E) – Extended CSS file

```

# -*- coding: utf-8 -*-
"""
flask
"""

A microframework based on Werkzeug. It's extensively documented
and follows best practice patterns.

:copyright: 2010 Pallets
:license: BSD-3-Clause

# utilities we import from Werkzeug and Jinja2 that are unused
# in the module but are exported as public interface.
from jinja2 import escape
from jinja2 import Markup
from werkzeug.exceptions import abort
from werkzeug.utils import redirect

from . import json
from .compat import json_available
from .app import Flask
from .app import Request
from .app import Response
from .blueprints import Blueprint

```

Fig 1.2(F) – Flask module

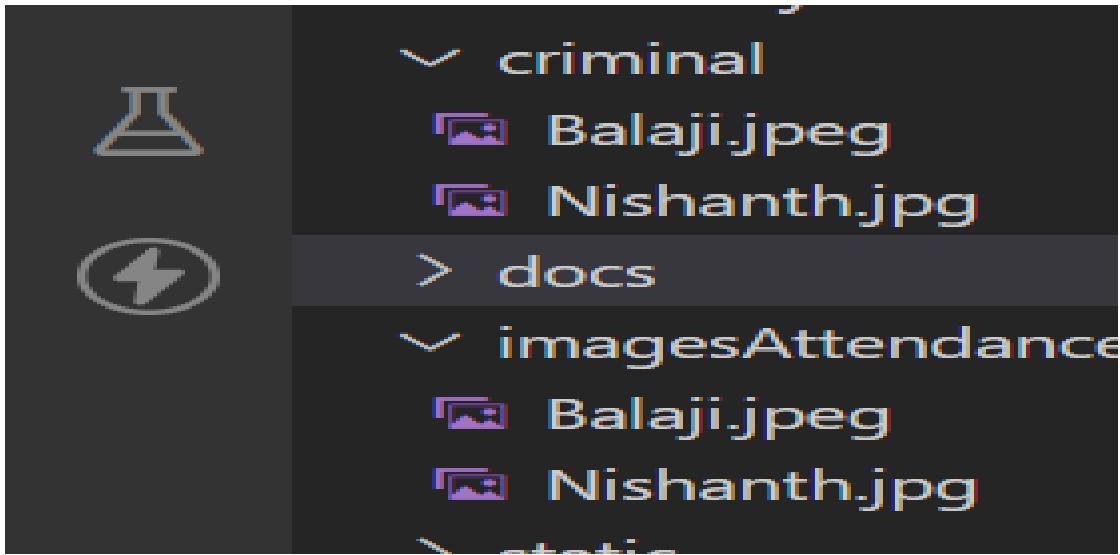


Fig 1.2(F) – Database for images

```

def __load_extra_py_code_for_module(base, name, enable_debug_print=False):
    module_name = "{}_{}".format(__name__, name)
    export_module_name = "{}_{}".format(base, name)
    native_module = sys.modules.pop(module_name, None)
    try:
        py_module = importlib.import_module(module_name)
    except ImportError as err:
        if enable_debug_print:
            print("Can't load Python code for module:", module_name,
                  ". Reason:", err)
        # Extension doesn't contain extra py code
        return False

    if not hasattr(base, name):
        setattr(sys.modules[base], name, py_module)
    sys.modules[export_module_name] = py_module
    # If it is C extension module it is already loaded by cv2 package
    if native_module:
        setattr(py_module, "__native", native_module)
        for k, v in filter(lambda kv: not hasattr(py_module, kv[0]),
                           native_module.__dict__.items()):
            native_module.__dict__[k] = v
    if enable_debug_print: print('  symbol({}): {} = {}'.format(name, k, v))

```

Fig 1.3(A) – Opencv module

The screenshot shows the PyCharm IDE interface with the Pyaudio module code open. The code is for the `get_format_from_width` function, which returns a PortAudio format constant based on the specified width. The code includes type annotations and exception handling for invalid widths.

```
    Returns the size (in bytes) for the specified sample *format*.

    :param format: A |PaSampleFormat| constant.
    :raises ValueError: on invalid specified 'format'.
    :rtype: integer
    """

    return pa.get_sample_size(format)

*def get_format_from_width(width, unsigned=True):
    """
    Returns a PortAudio format constant for the specified *width*.

    :param width: The desired sample width in bytes (1, 2, 3, or 4)
    :param unsigned: For 1 byte width, specifies signed or unsigned format.
    :raises ValueError: when invalid *width*
    :rtype: A |PaSampleFormat| constant
    """

    if width == 1:
        if unsigned:
```

Fig 1.3(B) – Pyaudio module

The screenshot shows the PyCharm IDE interface with the Jinja template code open. The code defines a template engine with various filters and environment settings. It imports modules like `MarkupSafe`, `BytecodeCache`, and `FileSystemBytecodeCache`.

```
# -*- coding: utf-8 -*-
"""
Jinja is a template engine written in pure Python. It provides a non-XML syntax that supports inline expressions and an optional sandboxed environment.
"""

from markupsafe import escape
from markupsafe import Markup

from .bccache import BytecodeCache
from .bccache import FileSystemBytecodeCache
from .bccache import MemcachedBytecodeCache
from .environment import Environment
from .environment import Template
from .exceptions import TemplateAssertionError
from .exceptions import TemplateError
from .exceptions import TemplateNotFound
from .exceptions import TemplateRuntimeError
from .exceptions import TemplatesNotFound
from .exceptions import TemplatesSyntaxError
from .exceptions import UndefinedError
from .filters import contextfilter
from .filters import environmentfilter
from .filters import evalcontextfilter
```

Fig 1.3(C) – Jinja template

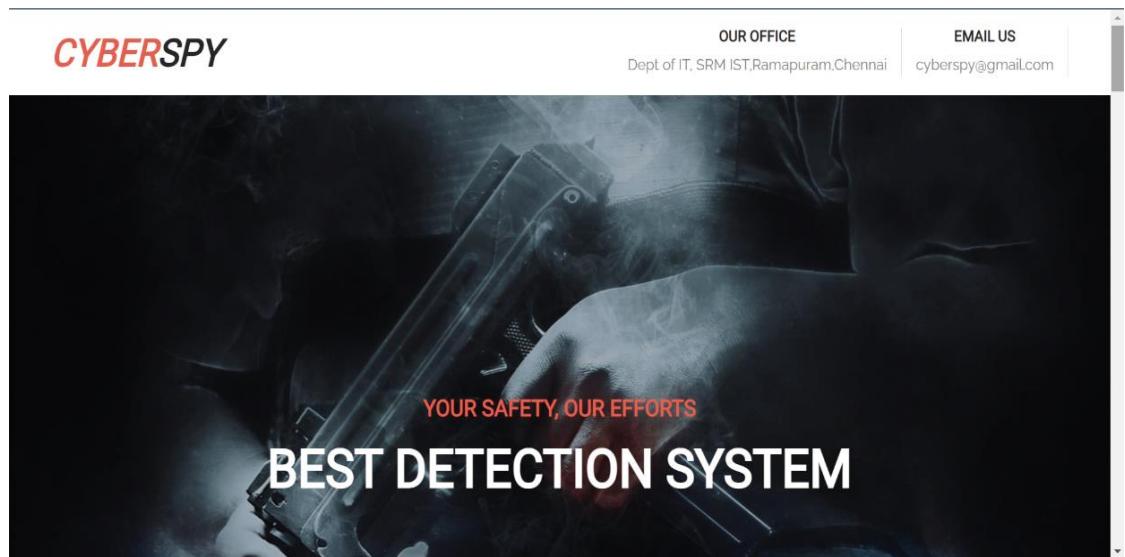
The screenshot shows the PyCharm IDE interface with the following details:

- Project View:** Shows the file structure of the `bmicalc` project, specifically navigating through the `site-packages\setuptools\pywin32_system32` directory.
- Code Editor:** Displays the source code for the `_init_.py` file under `pywin32_system32`. The code is part of the `distutils` module and defines the `Command` class and its `__init__` method.
- Toolbars and Menus:** Standard PyCharm menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help are visible at the top.
- Status Bar:** Shows the Python version (Python 3.9), the current file (`bmicalc`), and the date and time (18-09-2022).

```
162     class Command(_Command):
163         __doc__ = _Command.__doc__
164
165         command_consumes_arguments = False
166
167     def __init__(self, dist, **kw):
168         """
169             Construct the command for dist, updating
170             vars(self) with any keyword parameters.
171         """
172         _Command.__init__(self, dist)
173         vars(self).update(kw)
174
175     def _ensure_stringlike(self, option, what, default=None):
176         val = getattr(self, option)
177         if val is None:
178             setattr(self, option, default)
179             return default
180         elif not isinstance(val, str):
181             raise DistutilsOptionError("%r must be a %s (got %r)" %
182                                         (option, what, val))
183
184         return val
```

Fig 1.3 (D) – Win32API module

APPENDIX 2 – SAMPLE OUTPUT



CYBERSPY

OUR OFFICE
Dept of IT, SRM IST, Ramapuram, Chennai

EMAIL US
cyberspy@gmail.com

YOUR SAFETY, OUR EFFORTS

BEST DETECTION SYSTEM



PROFESSIONAL TEAM



We, IT Team from SRM University having team figure as 7. We are working in great spirit for your safety.

CUTTING EDGE TECHNOLOGY



Cyberspy is made of OpenCV 5.0 and AI with integration of ADABOOST algorithm using the mastermind python 5.0.

24/7 WORKING



Dont worry... Cyberspy will protect the nation and home



ABOUT US

4 YEARS EXPERIENCE

We are experienced in ML and we are full stack web developers developing cyberspy to reach heights.

CLICK THE BELOW IMAGES TO SEE THE APP IN ACTION

After clicking there will be a window popping in your start bar..open the window and use it...😊

Kindly note that the start of the application will take time to detect the camera..Be patient to witness the magic.😊😊

OUR APPLICATIONS

HIGH TECH SYSTEMS



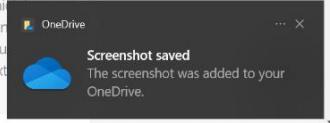
CRIMINAL DETECTION SYSTEM

Criminal detection system is connected to a criminals database which contains criminal images. It recognizes and detects faces. When it detects a face, it blares an alarm with a saying.



ROBBERY DETECTION SYSTEM

Robbery detection system is a system which is used to detect unknown faces. It is used in homes/offices and can be used in any industry. This system will sound an alarm with a text message if a face is found. Press Q to exit the application.



OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
... X

3
300
2
4.5



OUR TEAM



HAPPY CUSTOMERS



APPLICATIONS INSTALLED



RATINGS

OUR WARRIORS

MEET OUR TEAM

NISHANTH DAMODAR

Team Lead & Core Developer

Experienced in ML and a powerful full stack developer with exemplary knowledge in web building

in



BARATH BALAJI

Innovator & Ideator

Experienced in project management and leading the team in right path.

in



Buiting

in



in

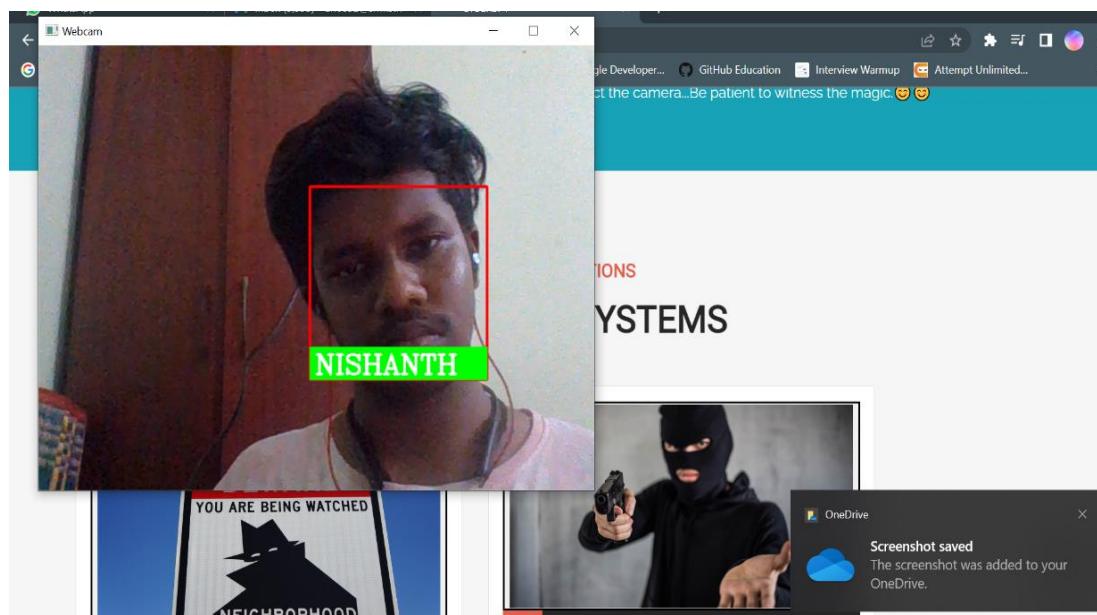
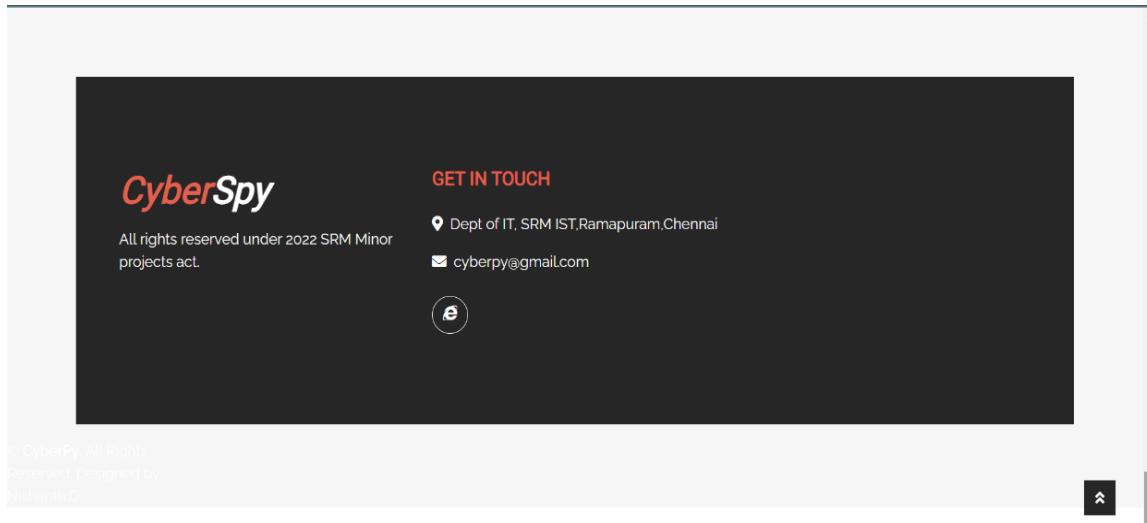
SUTHARSAN.V

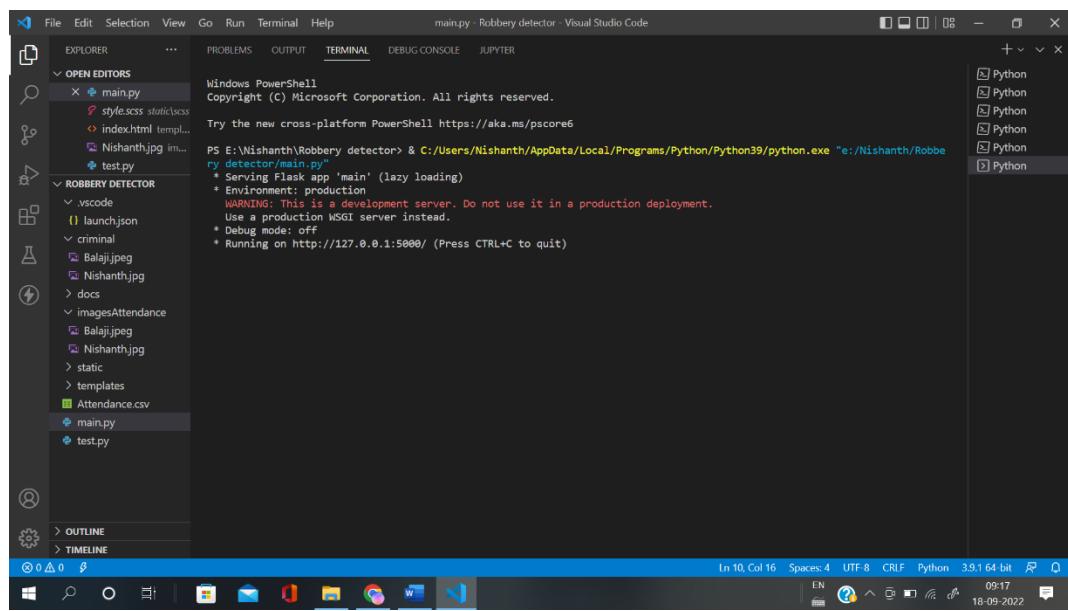
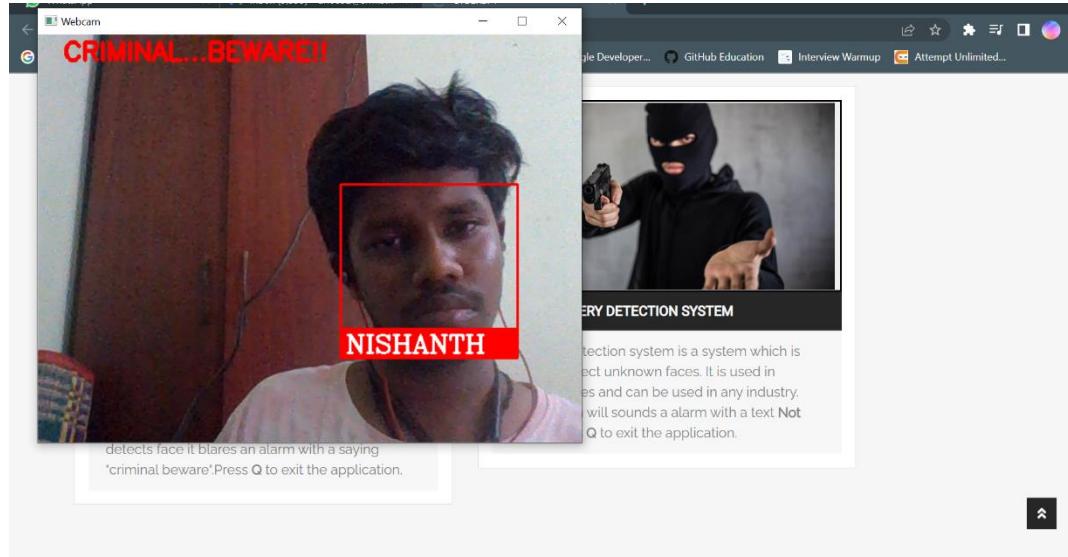
Testing Lead

Experienced in testing applications and developing reports for all the authority

in







The screenshot shows a Windows desktop environment with Visual Studio Code open. The terminal window displays a log of HTTP requests from a Flask application running on port 5000. The log includes timestamps, request methods, URLs, and response codes. The application serves static files like CSS, JS, and images, as well as dynamic routes like '/contact' and '/team'. The terminal interface includes tabs for PROBLEMS, OUTPUT, TERMINAL, DEBUG CONSOLE, and JUPYTER. The status bar at the bottom shows file paths, line and column counts, and other development settings.

```
File Edit Selection View Go Run Terminal Help main.py - Robbery detector - Visual Studio Code

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS E:\Nishanth\Robbery detector> & C:/Users/Nishanth/AppData/Local/Programs/Python/Python39/python.exe "e:/Nishanth/Robbery detector/main.py"
* Serving Flask app 'main' (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [18/Sep/2022 09:17:17] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:17] "GET /static/static/lib/waypoints waypoints.min.js HTTP/1.1" 404 -
127.0.0.1 - - [18/Sep/2022 09:17:17] "GET /static/lib/owlcarousel/assets/owl.carousel.min.css HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/css/style.css HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/lib/easing easing.min.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/lib/flaticon/font/flaticon.css HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/lib/jquerystrapValidation.min.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/mail/contact.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/js/main.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/lib/counterup/counterup.min.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/owlcarousel/owl.carousel.min.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/about.jpeg HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/service-1.jpg HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/team-3.jpeg HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/team-1.jpeg HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/team-2.jpeg HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/lib/flaticon/font/flaticon.woff2 HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [18/Sep/2022 09:17:21] "GET /static/img/favicon.ico HTTP/1.1" 404 -
```

This screenshot is nearly identical to the one above, showing the same terminal output and environment. The log of HTTP requests is identical, indicating no significant changes in the application's behavior or configuration between the two screenshots.

```
File Edit Selection View Go Run Terminal Help main.py - Robbery detector - Visual Studio Code

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS E:\Nishanth\Robbery detector> & C:/Users/Nishanth/AppData/Local/Programs/Python/Python39/python.exe "e:/Nishanth/Robbery detector/main.py"
* Serving Flask app 'main' (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [18/Sep/2022 09:17:17] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/lib/counterup/counterup.min.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/owlcarousel/owl.carousel.min.css HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/css/style.css HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/lib/easing easing.min.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/lib/flaticon/font/flaticon.css HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/lib/jquerystrapValidation.min.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/mail/contact.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/js/main.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/lib/counterup/counterup.min.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/owlcarousel/owl.carousel.min.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/about.jpeg HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/service-1.jpg HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/team-3.jpeg HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/team-1.jpeg HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/team-2.jpeg HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/lib/flaticon/font/flaticon.woff2 HTTP/1.1" 200 -
127.0.0.1 - - [18/Sep/2022 09:17:19] "GET /static/img/favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [18/Sep/2022 09:17:21] "GET /static/img/favicon.ico HTTP/1.1" 404 -
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Terminal Tab:** main.py - Robbery detector - Visual Studio Code
- Terminal Content:** A log of network requests from a Python application. The log includes timestamps, source IP (127.0.0.1), and various URLs requested. Some URLs are relative to the application's static files, such as 'main.js' and 'index.html'. Other URLs include 'jqRootstrapValidation.min.js' and 'jqBootstrapValidation.min.js'.
- Right Panel:** Shows a list of open Python terminals.
- Bottom Status Bar:** In 10, Col 16, Spaces: 4, UTF-8, CR/F, Python, 3.9.1 64-bit, 09:18, 18-09-2022.

REFERENCES

- [1] Akshay Rathod, Rushikesh Sawant – “Writing Overview on Criminal Recognizable proof in Mumbai utilizing DBSCAN “, Vol 12 No SUP 1 (2020): SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology.
- [2] Ashu Kumar, Amandeep Kaur – “Face Detection Techniques: A Review”, Artificial Intelligence Review 52(1) DOI:10.1007/s10462-018-9650-2
- [3] Xinghua Fan, Fuyou Zhang – “The system of face detection based on OpenCV”, 2012 24th Chinese Control and Decision Conference (CCDC), 2012, pp. 648-651, Doi: 10.1109/CCDC.2012.6242980.
- [4] Kruti Goyal, Kartikey Agarwal – “Face detection and tracking: Using OpenCV”, 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), 2017, pp. 474-478, Doi: 10.1109/ICECA.2017.8203730.
- [5] Mamata s.Kalas – “Real time face detection and tracking using opencv”, IRF International Conference, 5th & 6th February 2014, Pune India. ISBN: 978-93-82702-56-
- [6] Kushsairy Kadir, Mohd Khairi Kamaruddin – “A comparative study between LBP and Haar-like features for Face Detection using OpenCV”, 2014 4th International Conference on Engineering Technology and Technopreneurship (ICE2T), 2014, pp. 335-339, Doi: 10.1109/ICE2T.2014.7006273
- [7] Jian-Ming Liao – “Face detection based on adaboost”, 2009 International Conference on Apperceiving Computing and Intelligence Analysis, IEEE
- [8] S.V. Viraktamath, Aditya Khatakar – “Face Detection and Tracking using OpenCV”, The Standard International Journals (The SIJ), ISSN: 2321 – 2403
- [9] Dai Chen, Ji-lin Wang – “Face Detection Method Research and Implementation Based on Adaboost”, 2010 International Symposium on Intelligence Information Processing and Trusted Computing, 2010, pp. 643-646, doi: 10.1109/IPTC.2010.125.
- [10] H.W. Liu, G. Wang, and G.W. Zhang – “Research of AdaBoost Face Detection and OpenCV Implementation”, AMM 2014;651–653:482–5. <https://doi.org/10.4028/www.scientific.net/amm.651-653.482>.

[11] Rong Fu, Dan Wang – “University Classroom Attendance Based on Deep Learning”, 2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA), 2017, pp. 128-131, doi: 10.1109/ICICTA.2017.35.