Smart Security System using License Plate and Facial Recognition

Asad Ali Malik
Department of Electrical and Computer Engineering
COMSATS University
Islamabad, Pakistan
asadalimalik@comsats.edu.pk

Aamir Sharif
Department of Electrical and Computer Engineering
COMSATS University
Islamabad, Pakistan
aamirsharif19@gmail.com

Taimoor Ali Bhalli
Department of Electrical and Computer Engineering
COMSATS University
Islamabad, Pakistan
Taimoorbhalli004@gmail.com

Aiman Basharat Abbasi
Department of Electrical and Computer Engineering
COMSATS University
Islamabad, Pakistan
aimanbasharat99@gmail.com

***Abstract*— Threats have been increasing as the world is progressing. So, for that, security systems are being developed advanced day by day to tackle the rising security risks. To enhance security and safety in public places, this study offers a smart security system that integrates license plate and facial recognition technology. The technology takes pictures of license plates, analyses them using a dataset, and then takes pictures of people's faces, analyzing them using the same dataset when they enter and leave a building. Alerts are raised if there is a discrepancy between the dataset and the authorized cars and vehicles. Real-time monitoring and prompt reaction to security risks are provided by this technology. The system can be employed in a variety of locations, including parking lots, public safety surveillance, and building access management. The risk of security breaches is decreased, and security is improved by this approach.**

The method is based on the subsequent fundamental steps:

1. **A vehicle approaches the entrance gate, and the license plate recognition system captures an image of the license plate.**
2. **License plates will be detected and analyzed and will be checked in dataset.**
3. **If the license plate is recognized, then facial recognition is activated elsewise the vehicle will not be allowed to enter the premises.**
4. **The driver's face is captured and then is processed, analyzed, and then compared against the dataset.**
5. **If the face matches with the authorized individual, the gate opens, and then the vehicle is allowed to enter the premises elsewise not.**
6. **If either the license plate or facial recognition system detects an unauthorized vehicle or individual, security personnel are notified.**
7. **The system continuously monitors and records all vehicle and individual activities, creating a comprehensive record of all entries and exits.**

To achieve this, it will employ the basic principles of digital image processing, control systems and dataset management system. The significance of this system will be to make sure that the right staff members are entering the premises of the institution and exclude any intruders.

I. Introduction

The purpose of our project, a smart security system using license plate and facial recognition is to enhance security measures and improve surveillance capabilities in many public places such as university campuses, stations, malls, and airports.

By using license plate and facial recognition techniques and technologies, the system can automatically and efficiently detect and identify suspicious individuals and vehicles, and alert security. This enables security personnel to expeditiously respond to possible threats, stop criminal activities, and maintain the safety and security of the public and organization.

Additionally, this system can give security personnel crucial data and analytics/statistics for tracking and investigating the movements of people and vehicles, identifying patterns in suspicious behavior and activity, and creating reports for later review and investigations.

Overall, this technology has the potential to significantly increase the efficiency of security measures in public areas, creating a safe atmosphere for everybody.

The requirement for a sophisticated security system that can recognize and provide entry to only authorized people in a variety of situations, including gated communities, business buildings, and public spaces, is the issue this project attempts to solve. The inability of current security measures to distinguish between authorized and unauthorized persons could result in security breaches. This project intends to create a smart security system that can precisely identify and verify individuals before allowing access, assuring increased security measures, and lowering the risks of unauthorized access. It does this by merging license plate and facial recognition technologies.

The following are the goals of a smart security system utilizing facial and license plate recognition technology:

- Enhanced Security: This system's main goal is to increase security in public areas by precisely identifying and authenticating people and vehicles entering and leaving a facility. By doing so, you can lessen the chance of security lapses and guarantee everyone's and your property's protection.
- Real-Time Monitoring: The system should provide real-time monitoring capabilities to enable security personnel to quickly identify and respond to security threats.
- Reduction of Costs: The system should help reduce the need for human personnel, which can result in cost savings over the long term.
- Accurate Identification: The system should accurately identify and verify individuals and vehicles, using both license plate and facial recognition technologies, to minimize false positives and false negatives.
- Dataset Management: The system should have an efficient dataset management system to store and manage the data collected from the license plates and facial recognition technologies.
- Flexibility: The system should be flexible and adaptable to different types of public places and security scenarios, such as

parking lots, building access control, and public safety surveillance.

Overall, this system's goals are to deliver trustworthy, effective, and precise security solutions that improve public safety and reduce the chance of security breaches.

During this project we will design a system which involves integrating license plate and facial recognition technologies to create a smart security system that can accurately identify and verify individuals and vehicles. The project involves the selection and integration of appropriate hardware and software components to support the system, including cameras, sensors, and processors. The system requires the development of a dataset management system to store and manage the data collected from the license plate and facial recognition technologies. So, an array will be used to store all the authorized number plates. We will be developing and implementing digital image processing algorithms to analyse the images of license plates and faces captured by the cameras. Yolo v4 is the algorithm used to detect the license plate and OCR is used to read text from the plates. After that Haar cascade classifier to detect faces from the images and then fisher face algorithm is used for face recognition. After that testing and evaluation of the system will be done to ensure that it meets the objectives and functions as intended.

II. RELATED WORKS

The work done in the research paper cited at [1], presents an analytic approach using image processing techniques in the software MATLAB to implement a number plate recognition system. The simple methodology of this system involves segmentation and comparison of an image with a predefined dataset.

The utmost objective of research paper cited at [2] is to construct and implement a real time system for the monitoring of traffic rules and for law enforcement policy in smart surveillance systems, to avoid major crimes. The work depicts two of the You Only Look Once models together with the process of dispersion and detection of a license plate and their integration that deals with the administration and management of a parking lot.

The work done in the research paper cited at [3] is directly related to facial expression and emotion once the face is being detected. This paper overviews the details of this neural network algorithm, its limitation, performance, and the CNN architecture behind it.

The main motive of this research paper cited at [4] was to propose a solution for major security issues. So, it focuses on a system based on facial recognition, which can detect unauthorized individuals. In the software part it uses several algorithms e.g., gesture geometric model and corresponding face rotation models.

The work done in the research paper cited at [5] describes a system utilizes video processing techniques with a Raspberry Pi to detect and extract the number plate image from the vehicle's video footage. Various methods and algorithms are employed to extract the number from the captured images. This system finds applications at entrances of college gates and highly restricted areas.

The main objective of the research paper at [6] is the detection and recognition at the edge using optimization technique. Which introduces a two-stage pipeline processing for detection and recognition of face that includes various factors such as extraction of marks and information, face tracking, server receiving the accurate and best faces. For the licensee plated detection and recognition YOLO and LPRNet are used for reading and detection.

III. SOFTWARE DESIGN AND IMPLEMENTAION

So, the first part of our project is License Plate recognition. For that there are two main things to do in its recognition. First, we must localize it correctly and then read text from it accurately.

License plate recognition consists following steps:

• Image Capturing: An image of a vehicle’s license plate will be captured by a well-placed and balanced camera.

• Image Processing: Then the image is processed to detect the license plate area using Yolov4 algorithm which will be discussed extensively ahead.

• Character Segmentation: After localization of plate, the characters on it will be segmented using OCR algorithm. The individual characters are separated from each other.

• Character recognition: After segmentation, the characters are recognized using OCR technology.

• Dataset: After recognition of characters, it will be compared with the authorized licensed plates. If it’s matched, then we will move on next step which is face recognition and if it’s not then the entry is denied.

Other factors that are considered while designing LPR:

• Camera specifications and positioning: The camera is configured and positioned in such a way that it captures clear images of the license plates even in low light areas.

• Image processing algorithms: For accurate and efficient detection, segmentation of license plate the algorithms used must be of advanced level. So, there are very minimal chances of error because it will compromise security. So, we used yolov4 and OCR algorithms for license plates which are the most accurate and efficient algorithms out there.

• Dataset Management: The data is of very importance. So, it must be stored and managed properly. We initialized an array in our software for storing license plates and they are managed well.

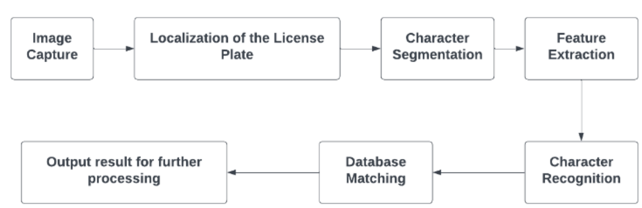

Fig 1 Block Diagram for Development of LPR System

After License Plate recognition, face detection and recognition process start. It consists of following steps:

• Image acquisition: The software captures images of faces to determine and analyze them. This is done with the external camera.

• Preprocessing: The captured images have some undesirable features like noise or lighting problems. Preprocessing techniques such as gray scaling, resizing, and scale factor to enhance the quality of the images.

Here are the general steps involved in facial recognition:

• Image capture: The system captures an image of a person's face driving the vehicle using a camera. The image is typically a high-resolution photograph.

- Face detection: Our system will use a computer vision algorithm to detect and locate faces within the captured image. In this step facial features such as the eyes, nose, and mouth are identified.
- Face recognition: Once a face has been detected, the system compares it to a dataset of authorized individuals.
- Verification: After the detected face is compared with the dataset, the returns a match score. If the score surpassed a certain threshold, the system contemplates the face an appropriate match and go ahead with the desired action which is to grant access if face is matches elsewise deny the entry of that vehicle and individual.

In general, the facial recognition part of our system is designed to provide an added layer of security beyond the license plate recognition system. By integrating both license plate and facial recognition, the system provides more effective security, safety, and access control.

There are four main algorithm/processes used in this project:

1. YOLOv4
2. OCR
3. Haar Cascade classifier (viola-jones algorithm)
4. Fisherface Recognizer

1. YOLOv4

YOLOv4 (You Only look once version 4) is a real time object detection model that can locate and detect objects in an image or real-time video. This version is famous for high accuracy and fast inference speed. This algorithm consists of a single neural network for the detection of objects in an image or video stream. It is trained on a large dataset of annotated images.

The architecture of YOLOv4 consists of a backbone, neck, and head networks. input image is used to extract features from the backbone, features are connected to backbone through neck, the detection of objects are done the head in an image.

The aforementioned backbone network of YOLOv4 is based on the CSPDarknet53 architecture, a modified version of the Darknet53 architecture used in YOLOv3.This architecture uses CSP (cross-stage partial connection) module that allows faster training and better feature extraction.

The neck network of YOLOv4 is called Spatial Pyramid Pooling and it combines features extracted by backbone at different scales. This SPP module captures features at different scales using multiple pooling layers with different kernel sizes.

The head network of YOLOv4 is composed of multiple detection heads that detect objects at different scales. Each detection head consists of a convolutional layer that predicts the bounding box coordinates and class probabilities for a set of predefined anchor boxes. The anchor boxes are used to define the shape and size of the objects that the algorithm is looking for.

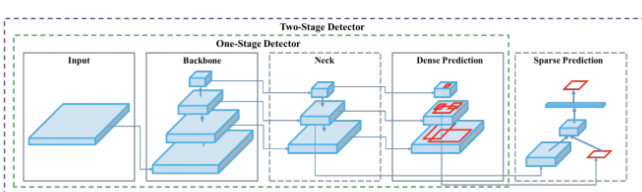

Fig2: Anatomy of Object Detector

The training process of YOLOv4 compromises of two main stages:

- Pretraining
- fine-tuning

During the pretraining stage, the backbone network is trained on a large dataset of labeled images. The pretraining stage is done to learn general features that can be used for a wide range of object detection tasks.

In the fine-tuning stage, the entire YOLOv4 network is trained on a smaller dataset of images that are specific to the task at hand. During fine-tuning, the weights of the backbone network are frozen, and only the weights of the detection heads are updated.

The inference process of YOLOv4 consists of passing an input image through the network and attaining the predicted bounding boxes and class probabilities for the objects in the image. The algorithm uses a non-maximum suppression (NMS) algorithm which is used to remove duplicate detections and select the most confident detections.

Non-maximum suppression (NMS) is a technique used in object detection algorithms to select a single entity out of many overlapping entities. It basically eliminates duplicate detections of the same object. After the object has been detected, the algorithm may detect multiple bounding boxes around the same object, with different degrees of confidence scores. To eliminate these multiple and duplicate detections, NMS is applied to the bounding boxes by selecting the box with the highest confidence score and cutting off all other boxes that overlap with it above a certain threshold.

The NMS algorithm first sorts the detected bounding boxes by their confidence scores. The highest score box is selected as the reference box, and all other overlapping boxes are compared based on their intersection over union (IoU) ratio. If the IoU ratio of a box with the reference box is above a certain threshold, it is suppressed and discarded. This process is repeated until all the boxes have been compared and the remaining boxes are treated as final detections.

NMS helps to remove duplicate detections and improve the accuracy of object detection algorithms by selecting only the most confident and non-overlapping bounding boxes.

In our project we set NMS threshold at 0.4 and it selects the most confident license plate accordingly.

Anchor boxes are a set of pre-defined bounding boxes of different shapes and sizes that are used by YOLOv4 to improve the accuracy of object localization. Anchor boxes predict the bounding box coordinates and class probabilities for each detected object.

During the training process, anchor boxes are assigned to each ground-truth object in the training dataset based on their size and position relative to the object. The predicted bounding box coordinates are then calculated based on the location and size of the assigned anchor box.

The YOLOv4 algorithm employs a method called k-means clustering to identify the most suitable anchor boxes for the given dataset. This process involves grouping the ground-truth bounding boxes based on their size and aspect ratio, and then selecting the cluster centroids as the anchor boxes.

By utilizing anchor boxes, object detection accuracy is enhanced greatly as the algorithm becomes better equipped to predict the precise size and shape of the objects accurately. This is particularly important for objects that vary greatly in size and shape, such as cars in a license plate recognition system.

Fig 3: License plate detection

2. OCR

Optical Character Recognition is a process that converts text from an image into a format that can be read by a machine. When you scan a document or receipt, it is saved as an image file, which cannot be edited or searched as text. However, using OCR technology, the image is converted into a text document, which makes it possible to edit, search, or count the words in the document as if it was a regular text file.

The OCR works by using the following steps:

- Image Acquisition: During image acquisition, scanners are used to read documents and convert them into binary data. Once the image is scanned, the OCR software analyzes it and distinguishes between the light and dark areas, classifying the former as the background and the latter as text.

- Preprocessing: To set up the image for text recognition, the OCR starts a cleaning process to eliminate any errors contained in the scanned image. This involves modifying the document's alignment and orientation, eliminating spots in digital image, or smoothing the edges of text images, and cleaning up boxes and lines in the image.

- Text Recognition: OCR software uses two methods, which are pattern matching and feature extraction, for text recognition.

- Pattern Matching: Pattern matching is a method that consists of obtaining a character image, known as a glyph, and comparing it with an already existing glyph with resembling scale and font.

- Feature Extraction: the OCR analyzes the characters by identifying and separating their features, such as lines, closed loops, line direction, and line intersections. Then OCR software then uses features to equate and match the character with the nearest neighbor among its multiple stored characters.

- Post preprocessing: Once the software has examined and extracted the text from the image, it transforms it into a digital format such as a text file or a word processor document.

Fig 4: Character reading from license plate

Our code uses OpenCV and Tesseract OCR to detect license plates in video frames. It also performs matching of detected license plate numbers with a set of pre-defined registered license plates.

The program captures the video frames from a connected camera and processes each frame using the licensePlateOCR() function. This function uses a pre-trained YOLOv4 object detection model to detect license plates in the input frame. It then crops the detected license plate regions, applies some image processing techniques like thresholding and contour detection to extract the text regions from the license plates, and then performs OCR using Tesseract to extract the text from the image.

The extracted text is then cleaned up, matched with the pre-defined registered license plates, and some action is taken based on the match result. The function returns the input frame with bounding boxes drawn around the detected license plate regions and the detected license plate number as a string.

3. Haar Cascade Classifier

Identifying or detecting a custom object in an image is object detection. This task can be accomplished using various techniques, but we used the haar cascade, which is the simplest method to achieve object detection.

It is an object detection algorithm which is based on machine learning that was proposed by Viola and Jones in their paper "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. This algorithm is extensively for detecting objects in images or videos which is very helpful in computer vision applications. This algorithm needs a lot of positive images of faces and negative images of non-faces to train a cascade of classifiers.

The algorithm can be described in four phases:

- Calculating Haar Features
- Creating Integral Images
- Using Adaboost
- Implementing Cascading Classifiers

The first step of this algorithm is to gather the Haar features. This Haar feature is substantial calculations that are done on nearby rectangular areas at a particular location in a detection window. To calculate the result, the process involves adding up the pixel intensities within each region and then determining the differences between these sums. Some examples of Haar features below:

A) Edge features
B) Line features
C) Four-rectangle features

These features make it easy to determine the edges or the lines in the image, or to select areas where there is a drastic change in the pixel intensities.

The above-mentioned features can be challenging to determine for a large image. This is where integral images are useful because by using these integral images, the number of operations. Integral images speed up the calculation of these features. Rather than computing at every pixel, it creates sub-rectangles and makes array references for each of those newly created sub-rectangles. These are then used to calculate the Haar features.

Adaboost substantially selects the best features and then trains the classifiers to utilize them. It uses the integration of “weak classifiers” to create a “strong classifier” that the algorithm uses to detect objects.

Weak learners are created by moving a window over the input image, and calculating Haar features for each subdivision of the image. This difference is compared to a learned threshold that splits non-objects from objects. As these are “weak classifiers,” many Haar features are required for accuracy to create a strong classifier.

The cascade classifier consists of a series of stages, where every stage is a collection of weak learners. Weak learners are trained using adaboost, which acknowledges a highly accurate classifier from the weak leaner’s mean prediction.

Depending on this prediction, the classifier chooses to indicate an object either was present (positive) or moves on to the next window region (negative). Stages are created to reject negative samples as quickly as possible, because most of the windows are irrelevant.

It is very critical to maximize a low false negative rate, because determining a face as a non-face will severely damage our object detection algorithm.

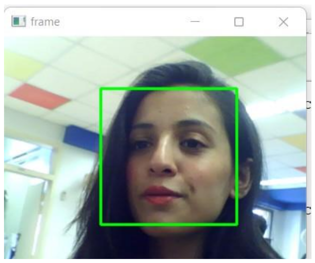

Fig 5: Face detection using haar cascade classifier

4. Fisherface Algorithm

Fisherfaces is a highly recognized and widely used algorithm in facial recognition that is considered more efficient and superior to other methods such as eigenface. The algorithm employs the use of Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to achieve accurate results.

Several key steps involved in face recognitions includes:

- capturing an image
- extracting features
- Comparing these features with defined dataset
- Determine whether it is a match or not!

Eigenfaces considers illumination to be an important feature in face recognition, but it may not be the most effective method. One way to improve this is by adjusting the eigenfaces to extract individual features rather than treating them as a group, which can prevent the features of one person from dominating another.

The algorithm aims to extract standard components that distinguish one individual from another, thereby preventing the features of one person from overpowering another's.

For image recognition, this algorithm uses PCA to reduce the dimensions of the face space, and then applies LDA to obtain characteristic features of the image, also known as Fisher Linear Discriminant (FDL).

LDA is used to determine a linear combination of features that effectively differentiate between two or more classes or objects, making it useful for dimensionality reduction before subsequent classification. It attempts to represent the differences between multiple classes of data.

This method may not capture illumination variations as obviously as the Eigenfaces method. The data is assumed to be uniformly distributed in each class. The goal is to maximize the ratio of between-class scatter matrix and within-class scatter matrix. This approach can produce good results even with varying illumination.

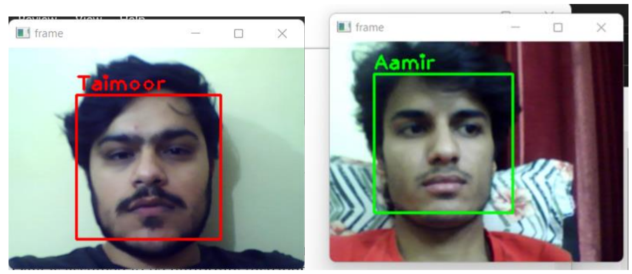

Fig 6: Recognition of face

Our code performs face recognition using the Fisherface method from OpenCV. Firstly, the detect_face function converts the input image to grayscale before applying the face detection algorithm because the OpenCV face detector algorithm works well on grayscale images.

After that, the predict function resizes the detected face image to a fixed size of 100x100 pixels. This is done to reduce the computational complexity and ensure that all the images are of the same.

Then Fisherface recognizer is trained on the input images using the train function. The train function takes in the preprocessed face images and their corresponding labels as inputs.

The predict function sets a confidence threshold of 180 for face recognition predictions. If the confidence score of a prediction is less than the threshold, the predicted label is inaccurate. Also, the prediction function draws a rectangle around the detected face using the coordinates obtained from the face detection algorithm. This helps in visualizing the detected face. Then the predict function draws the predicted label of the detected face on the image. This helps in identifying the person in the image.

IV. Hardware Implementation

After completing all the software part of our project, hardware is the next step to integrate it with the software and make our project more prominent. So, it will excellently show how our project will work.

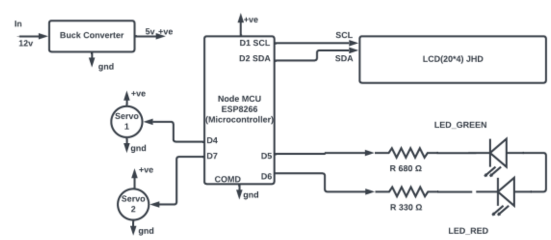

Fig 7: Block diagram of hardware

Our project consists of following components:

- Microcontroller (Node MCU ESP8266)
- LCD (20x4) JHD
- LCD I2C Module
- Servo Motors
- Buck Converter
- LED’s
- Resistors

. All the hardware components are connected to the microcontroller (Node MCU ESP82660, by the following steps:

o LCD is connected to the NodeMCU ESP8266 using the I2C protocol. This is done by connecting the SCL and SDL pins of the LCD to the corresponding pins on the microcontroller. The I2C library is used to communicate with the LCD and display the relevant data.

o Servo motors are connected to the microcontroller. For that we used PWM pins (D4 and D7) of the microcontroller, which produce the necessary signals to control the servo motors. The Servo library is used to control the motors and move them to the desired and specific position.

o Buck converter is connected to the NodeMCU ESP8266. This is done by connecting the buck converter's output to a GPIO pin on the microcontroller, which turns on or off the converter as needed.

o LED's and resistors to the microcontroller NodeMCU ESP8266. This is achieved by connecting the LED's to GPIO pins (D5 and D6) on the microcontroller, and suitable resistors are used to limit the current flow through the LED's. The digitalWrite function is used to turn the LED's on or off as needed.

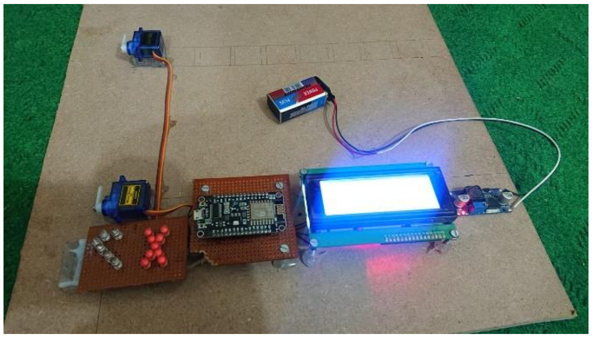

Fig 8: Hardware Implementation

After all the hardware components connection and configuration, the microcontroller is programmed to control its behavior. These servo motors act as gates for entry and exit points. It is programmed in such a way that the servo motors move as gate is opening when a recognized license plate and face is detected. After that its status, the license plate number, its time, and date are shown on the LCD. The program is written in C++ and uploaded on the microcontroller using the Arduino IDE.

V. TESTING AND RESULTS

After the complete integration of the system, we tested it on the dataset and checked its accuracy.

The license plate recognition system was tested on several photos of license place to determine its accuracy.

License plate	Recognition Accuracy
1	98.9
2	98.5
3	97

Table 1: License Plate accuracy

The facial recognition system was tested on several photos of an individual to determine its accuracy. A person was tested 25 times. When the photos are all tested with the proposed face recognition system, the data is computed to calculate the accuracy of the system.

From Table, it can be observed that the true and false recognition done by the proposed face recognition system. For the first person, 22 out of 25 recognitions are true. The true statement means that the identity of person in the photo that is recognized by the system is matched with real identity of the person. For the second person, 21 of his photos are recognized correctly and for the third one 24 out of 25 are detected correctly while for the last two people, 23 and 24 are recognized correctly. Thus, the accuracy of the system can be calculated.

Person 1	Person 2	Person 3	Person 4	Result
1	1	1	22	Person 4
21	0	0	4	Person 1
0	1	24	0	Person 3
1	23	0	1	Person 2
1	0	24	0	Person 5

Table 2: Facial Accuracy

Accuracy for first person: 88%

Accuracy for second person: 84%

Accuracy for third person: 96%

Accuracy for fourth person: 92%

Accuracy for fourth person: 96%

Overall accuracy: 91.2

VI. CONCLUSION

Our project could significantly improve security precautions in a variety of contexts. These cutting-edge technologies enable the system to precisely identify and validate people and vehicles, enabling regulated entry to restricted locations.

But for such a project, extensive technological know-how and resources would be needed, such as programming abilities, computer vision algorithms, and hardware elements like cameras and microcontrollers. Furthermore, while employing facial recognition technology, privacy issues must be taken into account.

Overall, the use of license plate and facial recognition in a smart security system has the potential to improve security measures, but it must be carefully designed and implemented with respect for both technological and ethical issues.

1) Limitations

Some possible limitations include:

- False positives: Incorrect license plate or facial recognition matches could cause the system to mistakenly designate someone as a threat, prompting pointless security warnings or interventions.

- Privacy issues: If the system is used in a public setting, there may be issues with the collecting and usage of personal information such facial images and license plate numbers.

- Cost: The price of the system's necessary hardware, software, and upkeep can be considerable, particularly for large-scale implementations.

- Maintenance: To guarantee that the cameras and software are operating correctly, the system needs frequent maintenance, which can be time-consuming and expensive.

- Environmental issues: The accuracy of the system may have some impacts by challenging-to-control environmental elements including weather and illumination.

2) Future Scope

- Integration with IoT Devices: building a completely automated/smart security system, the system might have connections with other IoT devices like smart locks, smart cameras, and smart lights.
- Cloud-based Storage: For quick access and retrieval, the recorded photos and videos in the cloud are stored in the system. This may ensure to manage and monitor the system remotely.
- Advanced Analytics: In detecting and notifying any suspicious behavior, aberrant activities, and potential threats, the system might be improved with many advanced analytics technologies. That might involve the detection of various objects, bodies in motion, and anomalies.
- Mobile Application: To allow the users to monitor and control the system remotely, a specific mobile application can be designed. That may include the following features like system configuration, notification and live video streaming.
- AI assistant integration: A compatible system with well-known AI assistants like Amazon Alexa, Siri and Google Assistant, which allows the user to operate the system using voice.
- Support for several Cameras: The system can be improved to accommodate numerous cameras for broad surveillance coverage. This may make it possible to monitor a huge area and better accuracy of the system.

REFERENCES

[1] Harshit Surya Koppisetti, A. M. (2021, July). Number Plate Recognition System using MATLAB. India. Retrieved from https://www.researchgate.net/publication/353234710_Number_Plate_Recognition_System_using_MATLAB.

[2] Ionuț-Andrei Pătrăuceanu, A. I. (2021). An Automatic Car License Plate Recognition System. Romania. Retrieved from https://www.researchgate.net/publication/356717707_An_Automatic_Car_License_Plate_Recognition_System.

[3] Salar Jamal Abdulhameed, A. M. (2023, March). Facial Expression Recognition Based on Deep Learning: An Overview. Retrieved from https://www.researchgate.net/publication/369870232_Facial_Expression_Recognition_Based_on_Deep_Learning_An_Overview.

[4] Michel Owayjan, A. D. (2013, December). Face Recognition Security System. America. Retrieved from https://www.researchgate.net/publication/259027363_Face_Recognition_Security_System.

[5] Chinmayi Gurav, R. G. (2019, apriil). A Review Paper on Vehicle Number Plate. Retrieved from https://www.ijert.org/research/a-review-paper-on-vehicle-number-plate-recognition-IJERTV8IS040246.pdf.

[6] Michel Owayjan, A. D. (2013, December). Face Recognition Security System. America. Retrieved from https://www.researchgate.net/publication/259027363_Face_Recognition_Security_System.

[7] "Face Recognition Using Fisherfaces." OpenGenus IQ: Computing Expertise & Legacy, 13 Oct. 2019, https://iq.opengenus.org/face-recognition-using-fisherfaces/.

[8] "What Is OCR? - Optical Character Recognition Explained - AWS." Amazon Web Services, Inc., https://aws.amazon.com/what-is/ocr/. Accessed 15 July 2023

[9] Mittal, Aditya. "Haar Cascades, Explained." Analytics Vidhya, 26 June 2021, https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d.

[10] NodeMCU ESP8266 Specifications, Overview and Setting Up. 15 Sept. 2021, https://www.make-it.ca/nodemcu-details-specifications/.

[11] "What Is OCR (Optical Character Recognition): How It Works & Application | Simplilearn." Simplilearn.Com, 16 Nov. 2021, https://www.simplilearn.com/what-is-ocr-optical-character-recognition-article.

[12] OCR Algorithms: A Complete Guide – Itransition. https://www.itransition.com/blog/ocr-algorithm. Accessed 15 July 2023.

[13] Jun 4, Jacob Solawetz, and 2020 10 Min Read □ □ □ □. "YOLOv4 - An Explanation of How It Works." Roboflow Blog, 4 June 2020, https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/.

[14] OpenCV 3 Object Detection : Face Detection Using Haar Cascade Classfiers - 2020. https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Object_Detection_Face_Detection_Haar_Cascade_Classifiers.php. Accessed 15 July 2023.

[15] "Medium." Medium, https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd0838210e57970d?gi=fcc95f8ae062. Accessed 15 July 2023.

[16] Jaiswal, Abhishek. "Object Detection Using Haar Cascade: OpenCV." Analytics Vidhya, 1 Apr. 2022, https://www.analyticsvidhya.com/blog/2022/04/object-detection-using-haar-cascade-opencv/.

[17] Anggo, Mustamin, and La Arapu. "Face Recognition Using Fisherface Method." Journal of Physics: Conference Series, vol. 1028, no. 1, June 2018, p. 012119. Institute of Physics, https://doi.org/10.1088/1742-6596/1028/1/012119.

[18] "NodeMCU ESP8266." Components101, https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet. Accessed 15 July 2023.

[19] apoorveinstein. "Interfacing 20x4 LCD With Arduino." Instructables, https://www.instructables.com/Interfacing-20x4-LCD-with-Arduino/. Accessed 15 July 2023..

[20] "In-Depth: Interfacing an I2C LCD with Arduino." Last Minute Engineers, 8 Nov. 2020, https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/.

[21] What Is a Servo Motor? - Understanding the Basics of Servo Motor Working. https://circuitdigest.com/article/servo-motor-working-and-basics. Accessed 15 July 2023.

[22] "What Is Resistor? - Symbol, Types, Unit, Applications, Color Code." BYJUS, https://byjus.com/physics/resistor/. Accessed 15 July 2023.

[23] "What Is LED? - Definition, Working, Properties, Uses, Advantages." BYJUS, https://byjus.com/physics/light-emitting-diode/. Accessed 15 July 2023.