Received Date: 22nd May, 2024
Revision Date: 5th June, 2024
Accepted Date: 22nd July, 2024

A Mobile Surveillance System with Face Detection and Recognition Abilities

Paras Pujara[1*], Prasamsha Dotel[2], Rajat Dulal[3], Suramya Pokharel[4], Suramya Sharma Dahal[5]

[1] *Dept of Electronics and Computer Engineering, Thapathali Campus, TU, E-mail:* paraspujara9825@gmail.com
[2] *Dept of Electronics and Computer Engineering, Thapathali Campus, TU, E-mail:* prasamshadotel652@gmail.com
[3] *Dept of Electronics and Computer Engineering, Thapathali Campus, TU, E-mail:* rjtdulal@gmail.com
[4] *Dept of Electronics and Computer Engineering, Thapathali Campus, TU, Email:* suramyapokharel7@gmail.com
[5] *Assoc Professor, Dept of Electronics, Communication & Information Engineering, Kathmandu Engineering College, E-mail:* suramya.sharma@kecktm.edu.np

Abstract— **This paper proposes a mobile surveillance system similar to an electronic dog which has the ability to move around a certain location all while detecting and recognizing the individuals and notifying the respective authorities about the detected person. The system uses an Arduino UNO microcontroller along with IR receiver and remote to function as a remote-controlled vehicle while also providing an option for automatic movement of the vehicle. The system also has an integrated camera that constantly feeds live video to a computer system. Haar-cascades take in the frames of videos and detect appearance of human face. Then, pretrained FaceNet Convolutional Neural Network (CNN) compares the detected face to the authorized faces in an attempt to recognize the face. The system also boasts the feature of email notification for which it employs the SMTP protocol to send an email with the detected face to the authorized personnel. The methods and techniques used to implement these features are presented in depth in this paper.**

Keywords— **Arduino UNO, Convolutional Neural Network (CNN), Electronic, FaceNet, Haar Cascades, IR receiver, SMTP protocol**

Introduction

As security systems continue to change, integrating cutting-edge technologies has become increasingly important. The need for advanced surveillance systems that can function independently and provide real-time tracking and alerting, all while being mobile and not stationary like CCTV cameras, to guard around a specific area, can be realized on proper contemplation. This paper proposes a similar system which integrates in itself almost all the functionalities that an actual watch dog can provide in the security domain. By combining robotic navigation with face detection and recognition capabilities, this paper aims to investigate the complete design and implementation of a robust surveillance system.

At heart of the mobilization of the surveillance system, lies an Arduino UNO [1] microcontroller which orchestrates the system's movements through an L293D [2] motor driver and TTL gear motors [3]. The microcontroller also plays a vital role in providing the system with two modes of operations – Manual and Automatic. In manual mode, a user must use a remote to control the movement of the system while in automatic mode, the system can move on its own at a designated path without a user's continuous involvement.

The system's face detection abilities are realized through a Haar Cascades [4] classifier especially adept in identifying front facing human faces in real time. In conjunction to the detection, the system's face recognition abilities derive from the state-of-the-art FaceNet Convolutional Neural Network [5]. The ability of FaceNet model to be able to generate meaningful embeddings from few images plays a vital role in authenticating different images of faces against a repository of known individuals. Finally, the automatic emailing mechanism present in the system helps alert the user of the system of the are currently under surveillance. This emailing mechanism follows the Simple Mail Transfer Protocol (SMTP) [6].

This paper aims to illuminate the interplay between hardware and AI components within the proposed surveillance system. It navigates through methodologies supporting the system's movement, along with the face detection and recognition, offering insights about the results along the way. By blending robotics with advanced image processing, this paper aims to shed light on a versatile and reliable surveillance system adaptable to various security needs.

Related Works

In their project named "The Electronic WatchDog" [7], the author proposed a simple security solution using IR sensors to detect intruders. At the entrance to the premises that needs to be secured, an infrared sensor transmitter and receiver were installed. When a person or intruder enters the door who is unaware of the security device installed at the entrance, the IR rays are cut. This cutting of IR rays sets off a chain of events in the circuit, culminating in the sounding of a burglar alarm. The owner of the premises may learn that someone has entered through the door after hearing the alarm. When the IR beam is interrupted, an alarm is

* *Corresponding Author*

triggered, alerting the property owner. While effective, this system lacks the versatility and advanced features provided by AI-driven solutions.

AIBO [8] Watchdog is a similar to the system proposed in this paper. AIBO robot has a built-in camera and other modalities, making this robot dog a viable alternative to static cameras. The AIBO watchdog project was created to protect the home environment. Evidences can be saved using its camera and microphone, and transferred to the appropriate person or instances immediately using its wireless connection. The developed AIBO watchdog is intelligent enough to notice related events and navigate in its home environment while patrolling. Furthermore, AIBO can protect itself from damage while patrolling and respond appropriately in most situations. To improve its intelligence, the AIBO watchdog was given the ability to prioritize events and use tools that it had previously encountered. The intelligence has been encapsulated in the AIBO watchdog's reasoning system.

In their paper [9], the authors of the Gabor-LeNet model, an enhancement of the LeNet-5 convolutional neural network, has demonstrated high accuracy on various face datasets. Likewise, Y.Zhiqi in his paper [10], introduced a model based on VGG-16 called MicroFace which improved the performance in face recognition compared to the base model. In the face detection domain, TinaFace [11], achieved an impressive 92.1% average precision (AP) on the hard test set with a single-model and single-scale approach. With test time augmentation (TTA), it further outperformed the state-of-the-art, achieving 92.4% AP. These results demonstrate the effectiveness of a straightforward yet robust baseline in face detection. This approach contrasts with other methods that introduce complexity through specialized techniques in model architecture, data augmentation, and label assignment, showing that simplicity can achieve competitive performance in the field of face detection.

Hardware System Methodology

A. Proposed Hardware Scheme

The hardware system revolves around an Arduino UNO [1] microcontroller. Arduino UNO needs 7.5 Volts of power supply which is fed using a pair of rechargeable batteries. Since the mechanism is one similar to a vehicle, TTL gear motors [3] have been used to control the wheels. The motors are connected to L293D [2] motor driver which in turn is connected to Arduino. The motor driver also requires the same 7.5 volts of power supply delivered by the battery pair. The system also has an attached IR receiver VS1838B [12] which, combined with the IR remote, helps in controlling the system's movement or changing the system modes between manual and automatic. The overall hardware architecture can be viewed in Fig. 1. The IR receiver is fed 5 volts of power supply by employing a voltage regulator on top of the battery pair.

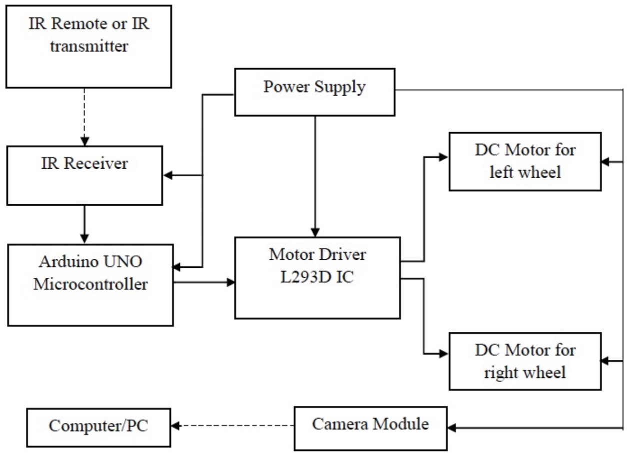

Fig. 1 Hardware Architecture of the System

B. Circuit Implementation

Arduino UNO, at its heart, contains ATmega328P [13] chip which operates at a clock speed of 16MHz [13]. The microcontroller can execute 20 Million Instructions Per Second (MIPS) and thus qualifies itself quite comfortably for the system's use case. The board of Arduino contains 14 digital input/output pins, 6 of which can be used for Pulse Width Modulation (PWM). Likewise, 6 analog input pins are available for analog sensors as well. Among the 14 digital pins, four pins are used for providing pulses to the L293D Motor driver and the fifth pin is used for receiving and decoding the IR signal.

The L293D is an integrated circuit having in it 16 pins with 8 pins on each individual side for motor control [c]. The IC is made up of 2 H-bridges [14] where each H-bridge can control one TTL gear motor. The connection established between the various components can be seen in Fig. 2.

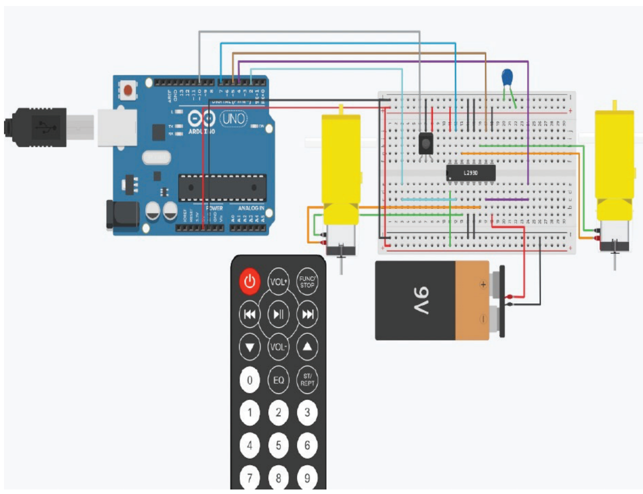

Fig. 2 Hardware Component Setup of the System

The IR receiver contains 3 pins – signal, ground and vcc [12]. The signal pin of the receiver is connected to one of Arduino's digital I/O pins while the remaining two are connected to their corresponding pins. The IR transmitter in the remote resembles a standard LED, other than the fact that the light it emits is in the infrared spectrum rather than the spectrum for visibility. A photodiode and a pre-amplifier in the IR receiver convert the infrared light into an electrical signal. The IR remote required installation of 'IRremote' [15] library to figure out the hexadecimal key codes of each button which came in handy in programming the Arduino. After making the connection as in Fig. 2, the code was uploaded using the USB connection. The detailed circuit diagram of the system can be seen in Fig. 3.

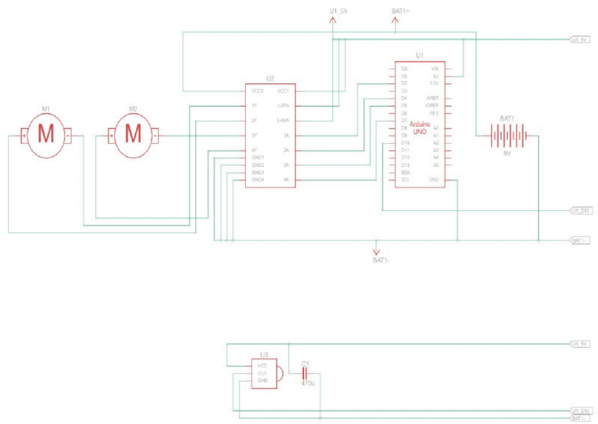

Fig. 3 Circuit Diagram of the System

C. IR Signal Modulation+

IR signal modulation is employed to avoid interference created from other sources of infrared light like sun or light bulbs. During this, a component in the IR remote called encoder changes a binary signal into its modulated electrical counterpart. The obtained modulated electrical signal is then sent to the transmitting LED which in turn converts this signal into a modulated IR light signal. Then, IR receiver simply demodulates and changes the IR light signal into binary prior to passing it to a module of microcontroller. Thus, the modulated IR signal is made up of a series of IR light pulses that are turned on and turned off at different times.

Every time a button is pressed on the remote, a code with unique hexadecimal value is generated. This is the same information which is then modulated and sent to the receiver via IR. For deciphering which key was pressed, the microcontroller contains the prior information, as described earlier, signifying which code corresponds to which button of the remote.

D. Modes of Operations

1) Manual Mode: Manual mode of operation makes it imperative on the user to control the system/vehicle's movement using the IR remote. Buttons on the remote have their designated functions. For instance, there is a button to turn the vehicles left which when pressed disables the motor of the left tire and only activates the right motor for a short while makes the right tire spin while the left tire acts as a pivot for rotating the vehicle leftwards. In the manual mode, one has to continuously press the corresponding buttons to make the system move as per need. Once operating the remote is stopped, the vehicle ceases its movement.

2) Automatic Mode: Automatic mode helps automate the movement of the system. First, the user has to specify the path for the vehicle to move. This is done by manually controlling the vehicle to move in the desired path. When a certain key is pressed during this manual control, equivalent hexadecimal code is generated and a counter is initiated. The value of the counter keeps on increasing based on time until next signal is received. As soon as the next signal is received, it stops the previous counter and initiates the next counter for the next direction, and so on. In such a way, all these patterns are recorded in the EEPROM of the Arduino and thus the path gets stored in terms of corresponding movements. Finally, the path is automatically reproduced simply by pressing the start button.

Face Detection and Recognition Mechanism

A. Haar Cascades

Haar Cascades [4] is a machine learning-based approach which has been used for face detection in the proposed system. The live video stream obtained from the camera set up in the vehicle goes through a layer of Haar cascades. Haar Cascades work by detecting Haar-like features.

Haar-like features are patterns that can be used to identify the presence of specific structures within an image, such as edges, lines, and rectangles. These features involve simple rectangular features that compute the difference in the sum of pixel intensities between adjacent regions. The basic idea is to capture the presence of certain structures like edges, lines, and changes in texture. These features can be seen in Fig. 4.

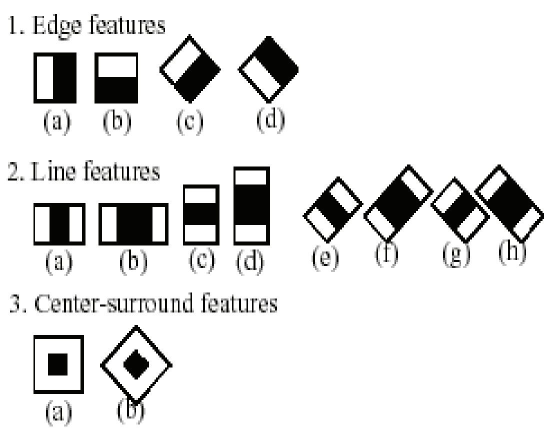

Fig. 4 Various Types of Haar Features [16]

For example, an edge feature might consist of two adjacent rectangles where the feature value is the difference between the sum of pixels in the left rectangle and the sum of pixels

$$f_{edge}(x,y) = \sum_{(i,j)\in R_1} I(i,j) - \sum_{(i,j)\in R_2} I(i,j) \quad (1)$$

Where, R_1 and R_2 are two adjacent rectangular regions, and $I(i,j)$ is the pixel intensity at position (i,j).

To efficiently compute these Haar-like features, the concept of an integral image (or summed-area table) is used. Integral image is a data structure used to quickly and efficiently compute the sum of pixel intensities in a given rectangular area.

$$S(x,y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i,j) \quad (2)$$

Where, $I(i,j)$ is the pixel intensity at position (i,j).

Using the integral image, the sum of pixel intensities within any rectangular region R with corners at (x_1,y_1) and (x_2,y_2) can be computed as:

$$Sum(R) = S(x_2,y_2) - S(x_2,y_1-1) - S(x_1-1,y_2) + S(x_1-1,y_1-1) \quad (3)$$

Ultimately to train the model, a large number of Haar-like features are computed for the training images. However, only a small subset of these features will be useful for detecting faces. AdaBoost is used to select the most effective features and to train the classifier. AdaBoost [17] iteratively selects the Haar-like feature that best classifies the training samples (with respect to the current weights). The selected weak classifier is then used to update the weights of the training samples, increasing the weights of the misclassified samples and decreasing the weights of the correctly classified samples. This process focuses the learning on the harder-to-classify samples. The final strong classifier is a weighted combination of the selected weak classifiers.

The trained strong classifiers are then organized into a cascade. The cascade is composed of multiple stages, each stage being a strong classifier trained with AdaBoost. Early stages use fewer and simpler features to quickly reject non-face regions. Later stages use more features to accurately classify face regions. During detection, an image region is passed through each stage of the cascade. If a region passes all stages, it is classified as a face. If it fails any stage, it is immediately rejected as not containing a face. If a region passes all stages, it is classified as a face. If it fails any stage, it is immediately rejected as not containing a face. This mechanism can be seen in Fig. 5.

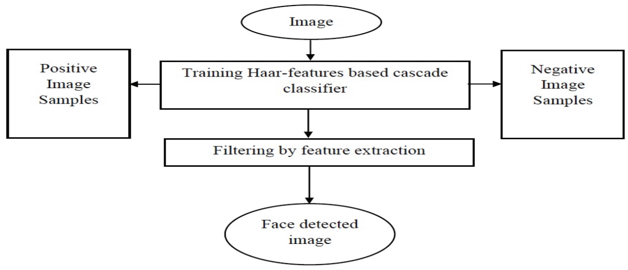

Fig. 5 Abstract Overview of Haar Cascades

The face detected by Haar cascades in any frame of the live video is cropped and extracted out to pass into upcoming mechanisms.

B. FaceNet CNN

For the face recognition part, FaceNet [5] model has been used. Unlike traditional face recognition systems that focus on extracting facial landmarks or specific features, FaceNet directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. This approach allows FaceNet to achieve state-of-the-art performance in face recognition task.

FaceNet uses a deep CNN [18] architecture to process input face images and extract features. The output of the CNN is a fixed-size vector of 128 dimensions, representing the face in a high-dimensional Euclidean space. This vector is known as an embedding. The key idea is that embeddings of faces of the same person should be close to each other, while embeddings of faces of different people should be far apart.

FaceNet is trained using a triplet loss [19] function. Each triplet consists of three images: an anchor image (A), a positive image (P) of the same person, and a negative image (N) of a different person. The triplet loss ensures that the distance between the anchor and the positive is smaller than the distance between the anchor and the negative by a margin $\alpha\alpha$:

$$\mathcal{L}(A,P,N) = max(\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha, 0) \quad (4)$$

Where, $f(x)$ is the embedding of image x and $\|\cdot\|_2$ denotes Euclidian distance.

A high-level overview of the face recognition process can be seen in Fig. 6. The cropped image obtained after Haar cascades is resized to 224 $\times$ 224 and the pixel values are normalized to a range of [0,1]. The image is then fed to the FaceNet model where the image is passed through multiple layers of convolutions, pooling [20], and fully connected layers [20] to produce the face embedding.

The resulting embedding is typically L2-normalized [21] to ensure it lies on the unit hypersphere. This means each embedding vector has a magnitude of 1, which simplifies distance calculations and improves numerical stability.

$$normalized\_embedding = \frac{embedding}{\|embedding\|_2} \quad (5)$$

In order to determine if the detected face is that of a known individual or not, the embedding of the detected face is compared with the embeddings of the known faces. The known face with the smallest distance to the query embedding is identified as the match. A threshold is set such that if the distance of the detected face is larger than the threshold value, the face is classified as unknown.

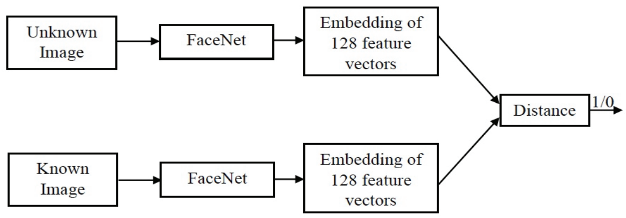

Fig. 6 Abstract Overview of Face Recognition Architecture

If the detected face is of a known individual, an email containing the image of the detected face is sent to the user with a message indicating the entry of a known individual in the system's proximity. Likewise, if the detected face is of an unknown individual, an email containing the detected face is sent but this time with a different message similar to an alert indicating the entry of an unknown individual in the system's proximity. Simple Mail Transfer Protocol (SMTP) [6] has been employed for sending these automated emails.

Results and Analysis

A. Hardware System Results

The L293D motor driver IC coupled with Arduino as the microcontroller worked fantastically well in controlling the TTL gear motors causing the vehicle to move smoothly. The IR receiver worked decently in all various types of lighting situations. The IR remote worked fine up until a distance of approximately 10 meters from the receiver. The buttons of the IR remote were in numeric format and were optimized as per Table 1.

TABLE I
IR REMOTE BUTTONS AND FUNCTIONS

IR remote buttons	Functions
2	Move Forward
4	Turn Left
6	Turn Right
8	Move Backward
5	Stop Movement
0	Repeat (For automatic mode)
1	Delete previously stored path and initiate recording of the new path

The movement patterns for memorizing the path for automatic mode were accurately recorded, and stored in the Arduino's EEPROM. When the '0' button of the IR remote is pressed, the system retraces the path previously stored. When '1' is pressed in the IR remote, the stored path gets flushed out of the EEPROM and new path can be entered for the system to trace. The functionality of the automatic mode was excellent for smooth surfaces however, on rough and/or rocky surfaces, the vehicle found it hard to track the exact path even though the movement patterns remained the same.

B. Haar Cascades Evaluation

To the implement the Haar Cascades, the publicly available pretrained 'haarcascade_frontalface_default.xml' file was taken from the 'OpenCV' github repository [22]. The accuracy of the model was evaluated in a custom created practical test dataset containing 78 images with face and 72 images with no faces in order to evaluate the performance in real world scenarios. Some sample images of the dataset can be seen in Fig. 7.

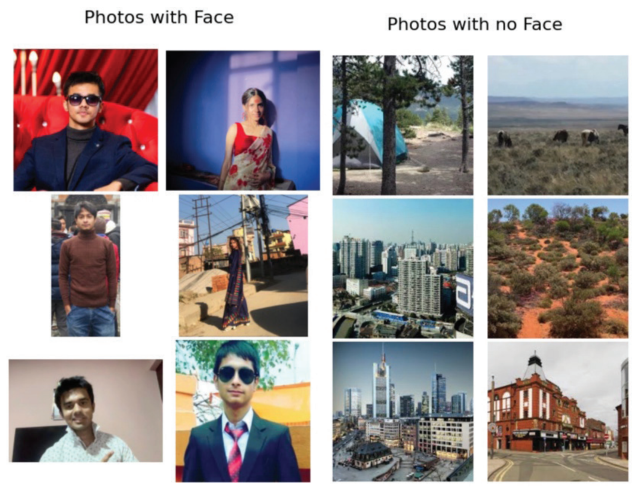

Fig. 7 Sample Images of Custom Created Test Dataset

The pretrained Haar Cascades model obtained an outstanding accuracy of 96.67% on the dataset, meaning it could with high accuracy detect faces in photos of various different types captured in variety of lightings. The classification report and confusion matrix can be seen in Table 2 and Fig. 8 respectively.

TABLE 2
CLASSIFICATION REPORT OF PRETRAINED HAAR CASCADES ON THE TEST DATASET

	Precision	Recall	F1-Score	Support
Face	0.95	0.99	0.97	78
No face	0.99	0.94	0.96	72
Accuracy			0.97	150
Macro avg	0.97	0.97	0.97	150
Weighted avg	0.97	0.97	0.97	150

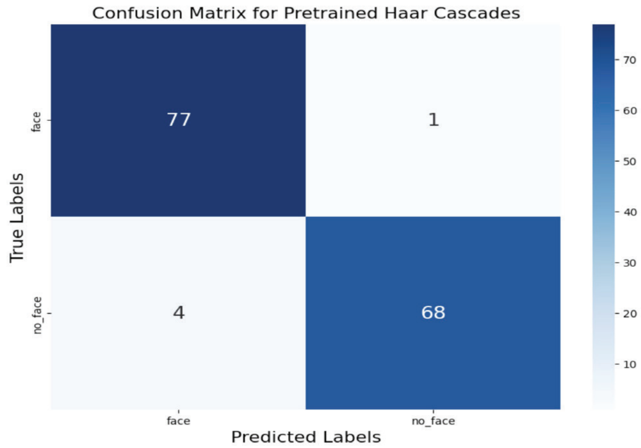

Fig. 8 Confusion Matrix of Pretrained Haar Cascades on Test Dataset

As evident from the classification report, the high precision for both 'face' and 'no_face' images suggests that most of the images that the model considered to have faces actually did have them and most images the model predicted to not have face did not have them. This is also very clear from the confusion matrix. Likewise, the high recall suggests that most of the images with faces were correctly identified as such and vice versa.

Also, as per the use case, the model was also clearly able to detect faces real time as can be seen in Fig. 9 where the green bounding box surrounds the facial area. This made Haar Cascades a reliable tool for face detection.

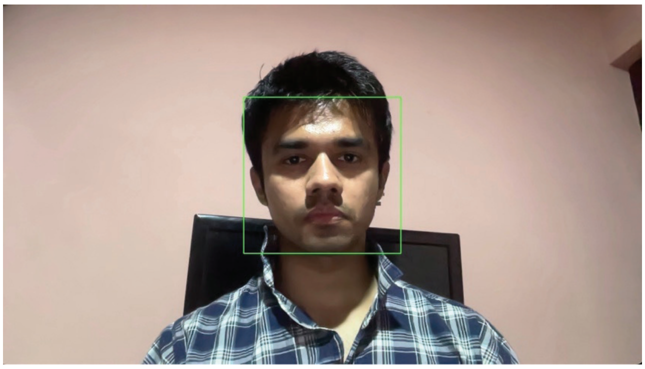

Fig. 9 Haar Cascades Detecting a Face in Real Time.

C. FaceNet Model Evaluation

Pretrained FaceNet model, loaded from the 'deepface' library, was used for face recognition purposes. The model was sent a custom-made practical test dataset containing a total of 78 images falling under 3 labels – 'Suramya', 'Prasamsha' and 'Rajat'. The model was supposed to classify each image into its correct label and if the model deemed it that a particular image falls not under any of the 3 labels, it predicts the image as 'Unknown'. Samples of the images in the dataset can be seen in Fig.10. The model was able to correctly identify the labels with 92.3% accuracy.

Fig. 10 Sample Images of Custom Created Test Dataset

The model was able to recognize faces of different individuals taken at various lightings and different conditions with relatively high accuracy. This makes the model fit the system's use case properly. The classification report can be seen in Table 3. The high precision values suggest that most of the images the model predicted to belong to a certain individual did actually belong to that particular individual. This is also evident from the confusion matrix as can be seen in Fig. 11. Likewise, high recall meant that the model successfully identified most of the actual positive instances for each label.

TABLE 3
CLASSIFICATION REPORT OF PRETRAINED FACENET MODEL ON THE TEST DATASET

	Precision	Recall	F1-Score	Support
Suramya	0.95	1.00	0.97	39
Prasamsha	1.00	0.78	0.88	27
Rajat	1.00	1.00	1.00	12
Unknown	0.00	0.00	0.00	0
Accuracy			0.92	78
Macro avg	0.74	0.69	0.71	78
Weighted avg	0.98	0.92	0.94	78

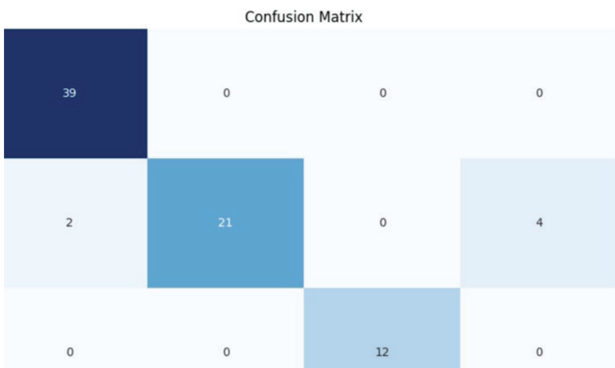

Fig. 11 Confusion Matrix of Pretrained FaceNet Model on Test Dataset

It is worth noting from the confusion matrix of Fig. 11 that quite a few images of the label ‘Prasamsha’ were classified incorrectly whereas all the images of ‘Suramya’ and ‘Rajat’ were classified accurately. This is approximated to have occurred because there are images falling under ‘Prasamsha’ category where face lies on drastically different and unique positions under extremely bright lighting conditions.

The model also was able to recognize faces real time, with low inference time, based on which the system could generate accurate responses as displayed in Fig. 12 and Fig. 13. The image on the left is the one captured by Haar Cascades while the image on the right (if exists) is of the known face, that the face in the captured image is deemed to be most similar to. Following this, an email, containing the captured face along with the status of the captured image indicating if the individual in the image is someone verified or not, is instantaneously sent to the user. The snapshot of email sent to the user can be seen in Fig. 14.

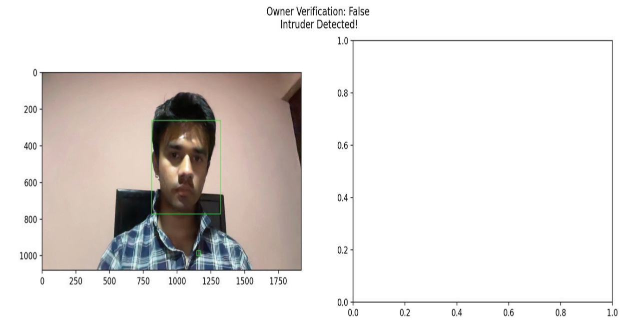

Fig. 12 FaceNet model correctly indicating the owner verification to be false based on the images of owner.

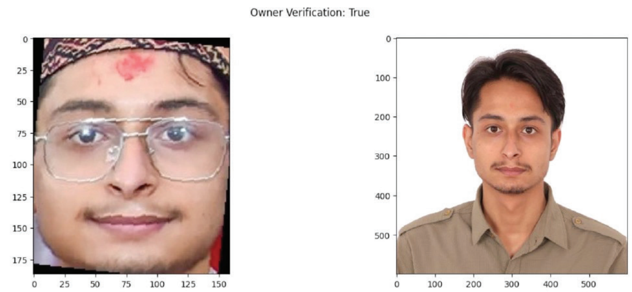

Fig. 13 FaceNet model correctly indicating the owner verification to be true based on the images of owner.

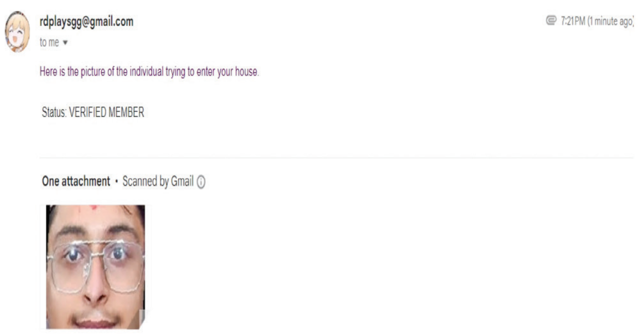

Fig. 14 Snapshot of Email Sent to the User

Conclusion

In this research, a robust movable surveillance system was successfully implemented. For the movement of the system, L239D motor driver along with TTL gear motors were used in conjunction with Arduino UNO microcontroller. The movement of the system was controlled using an IR remote that sent signals to IR transmitter, also fitted to Arduino microcontroller. The camera attached to the system would provide continuous live video feed on which Haar Cascades would detect faces when they appeared and FaceNet would try to recognize the faces that were detected. An automatic emailing capability was added to alert the user of the individual that appeared in the surveillance system’s viewpoint.

On a custom-made dataset, designed to resemble real-life scenarios, the models performed exceptionally well with Haar Cascades achieving 96.67% accuracy in the face detection domain and FaceNet achieving 92.3% on the face recognition side.

An issue was present in the automatic movement of the system in rough/rocky terrain where, despite the system making correct patterns of movement, failed to exactly retrace the specified path. However, this issue was not present in surfaces that were smooth and offered little friction.

In conclusion, the paper explored the designing of a robust and mobile surveillance system that coupled hardware and AI components in a versatile manner and provided insights about the results obtained. This paper can serve as a foundation in developing and enhancing more sophisticated security and surveillance systems.

Acknowledgement

We’d like to extend our heartfelt gratitude to our supervisor, Er. Suramya Sharma Dahal, for his invaluable guidance throughout this research project. Additionally, we are deeply grateful to the Department of Electronics and Computer Engineering at Thapathali Campus for their unwavering support.

References

[1] Arduino S.r.l. Arduino® UNO R3 Product Reference Manual. SKU: A000066. Modified May 20, 2024.

[2] Texas Instruments. “L293, L293D: Quadruple Half-H Drivers”. SLRS008D. September 1986, revised January 2016.

[3] Getriebebau NORD GmbH & Co. KG. “Motors M7000”. Bargteheide, Germany: Getriebebau NORD GmbH & Co. KG, n.d.

[4] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1, pp. I-I. Ieee, 2001.

[5] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815-823. 2015.

[6] Postel, Jon. *Simple mail transfer protocol*. No. rfc821. 1982.

[7] S. Vantepaka, "Electronic Watch Dog," Kakatiya Institute of Technology and Science, Warangal, 2021.

[8] L.J.M. Rothkrantz, H.J.A.M Geers, C.A.P.G. van der Mast. "AIBO as an aaaaIntelligent Robot Watchdog," Delft University of Technology, 2006

[9] Q. Dai, X. Luo, Z. Meng. "Research on face recognition based on Gabor-LeNet aaaaconvolutional neural network model," International Conference on Applied aaaaPhysics and Computing, ICAPC. 2020

[10] Y. Zhiqi. "Face recognition based on improved VGGNET convolutional neural aaaanetwork," IAEAC 2021. ISSN 2689-6621

[11] Y. Zhu, H. Cai, S. Zhang, C. Wang, and Y. Xiong, "TinaFace: Strong but Simple Baseline for Face Detection," arXiv preprint arXiv:2011.13183, November 2020.

[12] Vishay Semiconductors. "TSOP382.., TSOP384..: IR Receiver Modules for Remote Control Systems". Document No. 81733, Rev. 1.3. January 29, 2009.

[13] Microchip Technology Inc. "ATmega328P." Microchip. Accessed May 20, 2024.

[14] STMicroelectronics. "L298 Dual Full-Bridge Driver." Datasheet, rev. 5, October 2023.

[15] Arduino-IRremote Developers. "Arduino-IRremote." *GitHub*. Last modified May 2024.

[16] Chhajro, M. Ameen, Kamlesh Kumar, M. Malook Rind, Aftab Ahmed Shaikh, Haque Nawaz, and Rafaqat Hussain Arain. "Pedestrian Detection Approach for Driver Assisted System using Haar based Cascade Classifiers." *International Journal of Advanced Computer Science and Applications* 9, no. 6 (2018).

[17] Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." *Journal of computer and system sciences* 55, no. 1 (1997): 119-139.

[18] LeCun, Yann, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. "Backpropagation applied to handwritten zip code recognition." *Neural computation* 1, no. 4 (1989): 541-551.

[19] Vlb De Mel. "Triplet Loss." ResearchGate. Last modified December 2021.

[20] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86, no. 11 (1998): 2278-2324.

[21] Krogh, Anders, and John Hertz. "A simple weight decay can improve generalization." *Advances in neural information processing systems* 4 (1991).

[22] OpenCV. "OpenCV/haarcascades." GitHub. Last modified May 20, 2024.