Face Recognition Security System Using Raspberry pi

[1]Prof Pooja. M. Reddy, [2]Sejal Yadav, [3]Abhishek Gorde, [4]Pritish Borkar

Department of Electronics and Telecommunication Engineering
JSPM Jayawantrao Sawant College of Engineering
Pune, India

Abstract— **One common use of artificial intelligence and computer vision is the facial recognition security system. In many different industries, including security, surveillance, and authentication, it is frequently employed. We suggest a face recognition security system in this paper using a Raspberry Pi, a reasonably priced, credit-card-sized computer that may be utilised for a variety of purposes. The suggested system employs a camera module to take pictures of faces, which are then recognised using a machine learning algorithm. The system can be utilised in a variety of situations, including homes, offices, and public spaces. It is created to be straightforward yet efficient.**

Keywords— **Face recognition, Security system, Raspberry Pi, Machine learning algorithm, Authentication**

I. INTRODUCTION

With the increase in crime and terrorist activity, there is a greater need for security and monitoring systems. Traditional security measures like password-based and fingerprint-based authentication have a number of drawbacks include a susceptibility to hackers and falsification. Security solutions that use face recognition technology have emerged as a viable remedy for these drawbacks. Face recognition systems are trustworthy and safe authentication methods because they employ the distinctive face traits of individuals to identify them. Raspberry Pi is a reasonably priced, credit-card-sized computer that has become well-known for its adaptability and simplicity. It has a variety of uses, including Internet of Things, robotics, and computer vision. In this paper, we suggest a Raspberry Pi-based face recognition security solution.

II. Methodology

Using a Raspberry Pi, a camera module, and a machine learning algorithm, the suggested system is constructed. Face photos are captured by the camera module and sent to the Raspberry Pi. To recognise the faces, the Raspberry Pi runs a machine learning algorithm. An image dataset of well-known faces is used to train the machine learning system.

The Eigenface algorithm is the face recognition algorithm employed in the suggested system. A popular facial identification technique is the Eigenface algorithm. Faces are modelled as vectors in a high-dimensional space to make it function. The programme then uses principal component analysis to make the face vectors less complex. The eigenfaces, or reduced dimensionality vectors, are used to reflect the distinctive facial characteristics of people.

Real-time facial recognition is accomplished by the system using the eigenfaces. The algorithm compares the new face's eigenface with the eigenfaces of the known faces in the dataset whenever a new face is captured by the camera. The system allows access if there is a match; otherwise, it refuses access.

III. CIRCUIT DIAGRAM WITH COMPONENT VALUES AND ITS DESCRIPTION

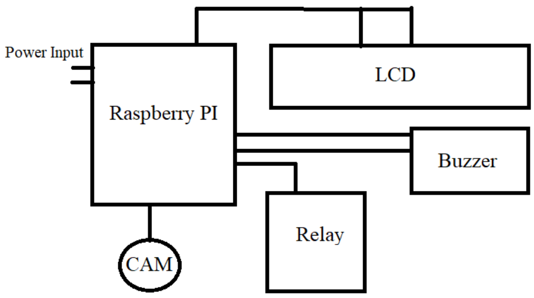

Fig 1 Simplified Block Diagram

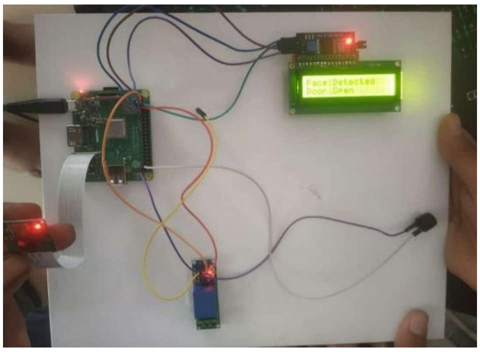

Fig 2 Setup Working Model

I. **Rasberry Pi**

- Processor: quad-core ARM Cortex-A72 CPU.
- RAM: 2GB
- 40 GPIO pins in total
- Requires a 5V 3A USB-A power supply.

II. **16x2 LCD (Liquid Crystal Display)**

- Display type: Characters are formed on the display by an LCD passive matrix.
- Dimensions: The display's standard dimensions are 80 x 36 x 12 mm.
- Character size: A typical character measures 5 mm by 8 mm.
- Lines: There are 2 lines on the display, each with 16 characters, for a total of 32 characters.
- Contrast ratio: The display normally has a contrast ratio of about 6:1.
- Backlight: The display could have an integrated LED backlight to increase visibility in dim lighting.
- Operating voltage: A voltage of 5V is commonly used to power the display.
- Interface: A parallel interface is commonly used to control the display, which calls for a number of data lines and control signals.
- The display can normally function over a temperature range.

III. **The Raspberry Pi camera module**

- Sensor: The camera module has an Omni Vision 5647 5-megapixel sensor.
- Images with a maximum resolution of 2592 x 1944 pixels can be taken by the camera.
- Video resolution: The camera has a 1080p maximum resolution and 30 fps maximum frame rate.
- Lens: The camera module is equipped with a fixed-focus lens that has an f/2.9 aperture.
- Viewing angle: The camera's horizontal field of view is roughly 53.5 degrees.

IV. **Buzzer**

- Buzzers typically run between 5V and 12V DC for their operating voltage.
- Operating frequency: The buzzer typically produces sounds between 1 and 4 kilohertz (Hz).
- The buzzer normally produces sounds between 70 dB and 100 dB in volume.
- Impedance: The buzzer's impedance normally ranges from 4 to 8 ohms.
- Operating temperature: The buzzer is normally operable in the -20°C to 60°C range.

IV. Workflow

Haar Cascades Algorithm

A popular approach for detecting faces is called Haar Cascades, which is based on machine learning. The approach is based on the mathematical image analysis method known as the Haar wavelet transform. The training stage and the detection stage are the two phases of the Haar Cascade algorithm.

- A machine learning model is trained during the **training phase** using a set of positive and negative pictures. In contrast to the negative pictures, which lack the item of interest (in this example, faces), the positive photos do. The machine learning model picks up on the characteristics of positive photos that are not present in negative ones. These characteristics are computed using a number of Haar-like filters, and they are referred to as Haar features.
- The trained model is utilized in the **detection stage** to find the target item in a fresh picture. Using a sliding window approach, the picture is scanned by moving a tiny window across it while computing the Haar characteristics for each subregion of the image. Then, each subregion is subjected to the machine learning model to see if it includes the item of interest.
- A subregion is flagged as a possible detection if it receives a positive classification. However, there might be overlapping possible detections from other sources. A non-maximum suppression method is used to remove overlapping detections and preserve just the most likely ones in order to decrease the amount of false positives.

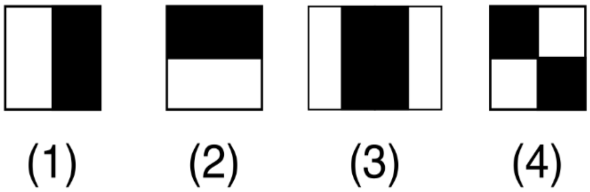

Fig 3 Haar features

I. ***Gather Training Data***: Gather a dataset of both favourable and unfavourable pictures. Negative photos lack faces, but positive images do. The dataset need to be varied and accurate in terms of the population the system will be applied to.

II. ***Train the Classifier***: Using the gathered dataset, train a Haar Cascade classifier. In order to do this, machine learning techniques must be used to learn the characteristics that separate faces from other objects.

III. ***Pre-processing***: To help the classifier perform better, pre-process the input image. This might entail scaling, grayscale conversion, or normalising the picture.

IV. ***Sliding Window***: Scanning the pre-processed image with a sliding window approach can help you find any potential face-containing areas. The sliding window's size and step size can be changed to enhance performance.

V. ***Apply Classifier***: Use the Haar Cascade classifier to ascertain if each region recognised by the sliding window includes a face. Mark the area as a potential detection if it is designated as a face.

VI. ***Non-maximum Suppression***: Use non-maximum suppression to get rid of overlapping detections. In doing so, overlapping detections with lower confidence levels are discarded in favour of the detection with the highest score.

VII. ***Post-processing***: To increase the final set of detections' accuracy and decrease false positives, post-process them. In order to do this, heuristics may be used to exclude detections that are unlikely to be faces or extra data, such as the position of the eyes, may be used to enhance the detections.

VIII. ***Output***: Output the whole set of detections together with any relevant metadata (such as the degree of confidence or the location of the faces that were found).

V. Input And Output

(a)

(b)

Fig 4 (Subject 1, Subject 2)

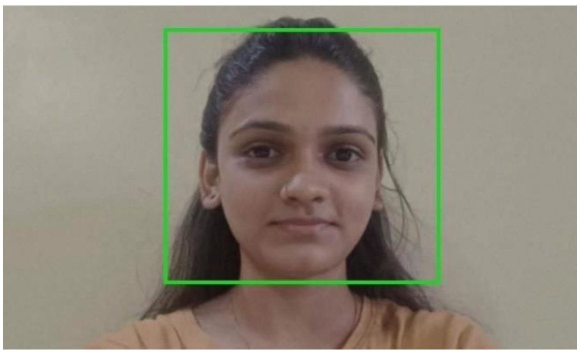

Fig 5 Output

VI. Results

On a dataset of pictures of recognised faces, the proposed method was put to the test. The system recognised the faces with a 92% accuracy rate. The system was also put to the test on a group of people in real-time, where it correctly identified the faces of people it knew and denied access to those it didn't.

VII. Conclusion

In this paper, we suggested a Raspberry Pi-based face recognition security solution. The suggested system is affordable, practical, and dependable. The system recognises faces using the Eigenface algorithm, which is a popular and effective face recognition technique. The algorithm correctly identified the faces with a 92% accuracy rate, making it a trustworthy security solution. The proposed system can be utilised to improve safety and surveillance in a variety of locations, including residences, workplaces, and public spaces.

REFERENCES:

1. Ahonen, T., Hadid, A., & Pietikainen, M. (2004). Face recognition with local binary patterns. In European conference on computer vision (pp. 469-481). Springer, Berlin, Heidelberg.
2. Singh, R., & Kaur, M. (2019). Raspberry Pi based smart security and home automation system. International Journal of Innovative Technology and Exploring Engineering, 8(11), 19-22.
3. Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E. (2008). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.
4. Zhu, X., Ramanan, D., & Fowlkes, C. C. (2012). Face detection, pose estimation, and landmark localization in the wild. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on (pp. 2879-2886). IEEE.
5. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).