IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Raspberry Pi-Based Deep Learning Face Recognition Security System

[1]Shafia Firdose, [2]Shiva Kumar B K, [3]Dinesh G

[1]Lecturer, [2]Lecturer, [3]Lecturer

[1, 2, 3] Department of Electronics and Communication,

[1]Government Polytechnic Mulbagal, [2]Government Polytechnic Hosadurga, Government Polytechnic Mirle, Karnataka, India

Abstract:

In today's rapidly evolving digital era, the need for intelligent and contactless security solutions has become increasingly significant. This project presents a Raspberry Pi-based deep learning face recognition security system designed to enhance door access control through biometric authentication. The system utilizes a Raspberry Pi board as the core processing unit, paired with a Raspberry Pi Camera Module to capture real-time facial images. These images are analyzed using a pre-trained Convolutional Neural Network (CNN) model built with TensorFlow and Keras, ensuring accurate and efficient facial recognition.

Upon successful identification of an authorized individual, the Raspberry Pi activates a relay module connected to a solenoid door lock, allowing secure and automatic door unlocking. The hardware setup is powered by a stable regulated power supply, ensuring continuous and reliable operation. The system eliminates the need for physical keys or access cards, thereby reducing the risk of unauthorized access or loss of credentials.

By integrating low-cost hardware components with advanced deep learning algorithms, this project delivers a smart, scalable, and real-time face recognition access control solution suitable for homes, offices, and institutional security. The system showcases the potential of edge computing in IoT-enabled environments, combining security, automation, and ease of use in a compact framework.

***Index Terms* -** Raspberry Pi, Face Recognition, Deep Learning, Convolutional Neural Network (CNN), Solenoid Lock, Relay Module, Raspberry Pi Camera, Biometric Authentication, Smart Door Lock, Real-Time Processing, IoT Security System, OpenCV, TensorFlow, Keras, Home Automation.

I. Introduction

In the modern digital age, the demand for intelligent and contactless security systems has grown significantly due to increasing concerns about safety, privacy, and convenience. Traditional security methods such as keys, access cards, and PIN codes are prone to being lost, stolen, or duplicated, thereby compromising security. Biometric authentication, particularly facial recognition, has emerged as a more secure and user-friendly alternative for personal and institutional security systems.

Facial recognition systems use unique facial features to identify or verify an individual, making them difficult to forge or bypass. With the rise of edge computing and Internet of Things (IoT), it is now possible to implement such systems on compact, low-cost hardware platforms like the Raspberry Pi. Raspberry Pi offers a powerful and affordable solution for deploying intelligent systems that can operate independently without the need for expensive computing infrastructure.

This project presents the design and implementation of a face recognition-based smart security system using Raspberry Pi, deep learning algorithms, and real-time image processing. The system utilizes a Raspberry Pi Camera Module to capture live facial images, which are then processed using a Convolutional Neural Network (CNN) model trained for facial recognition. When an authorized face is detected, the system activates a relay module that controls a solenoid door lock, granting access to the individual. The integration of these components creates a seamless, automated, and secure access control mechanism.

This system is particularly suitable for smart homes, office environments, and secure institutional settings. It eliminates the need for physical keys and enhances the overall security infrastructure with the power of artificial intelligence and embedded systems. Additionally, the compact size and low power consumption of the Raspberry Pi make it ideal for continuous deployment in real-world environments.

II. LITRATURE REVIEW

Face Recognition Techniques

Face recognition is a widely researched biometric identification method that uses facial features to recognize or verify individuals. Early techniques involved geometric feature-based methods, where key landmarks such as eyes, nose, and mouth were measured and compared. Template matching and eigenface approaches based on Principal Component Analysis (PCA) were later introduced to improve recognition accuracy. More recent classical methods include Fisherfaces, Local Binary Patterns Histogram (LBPH), and Support Vector Machines (SVM). Although these methods provided decent performance in controlled environments, they lacked robustness in varying lighting conditions, facial orientations, and real-world scenarios.

Deep Learning Approaches

The introduction of deep learning has revolutionized face recognition accuracy and efficiency. Convolutional Neural Networks (CNNs) have become the backbone of modern facial recognition systems due to their ability to learn hierarchical feature representations. Architectures such as FaceNet, DeepFace, VGGFace, and OpenFace have significantly improved performance on large-scale datasets. These models utilize embedding techniques to represent faces in a high-dimensional space, allowing precise classification and verification. Deep learning methods also offer better adaptability to diverse datasets and real-time performance when optimized for lightweight deployment.

Applications Using Raspberry Pi

The Raspberry Pi, a low-cost and compact single-board computer, has become a preferred platform for implementing IoT and AI-based applications. In facial recognition, it has been widely used for smart door locks, surveillance systems, attendance monitoring, and robotics. Several studies have integrated Raspberry Pi with cameras and relay modules to develop contactless security systems. Its GPIO pins allow seamless control of actuators like solenoids and relays, making it ideal for real-time automation. Despite its limited computational resources, the Raspberry Pi can handle deep learning inference through model optimization or the use of lightweight architectures such as MobileNet.

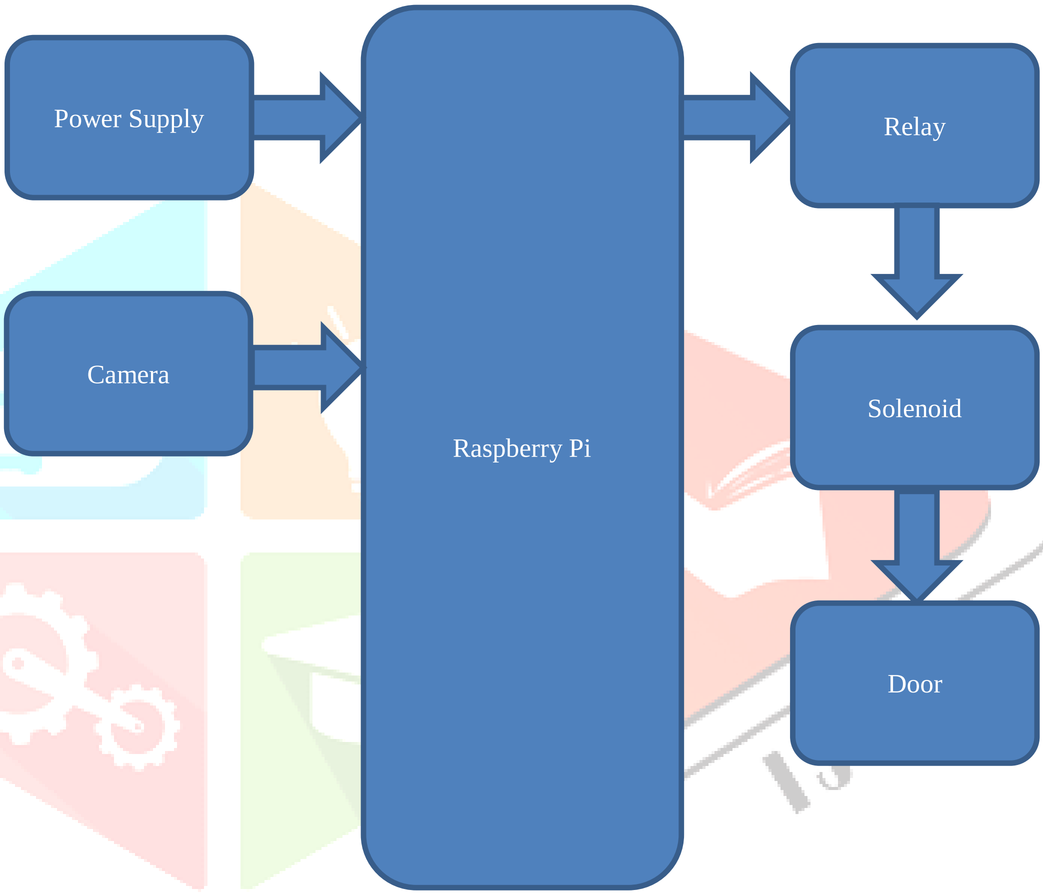
Research Gaps Identified

Although substantial work has been done in face recognition and smart access systems, several research gaps remain. Many existing systems depend on cloud-based processing, raising concerns about data privacy and latency. Others require expensive hardware or lack portability. Few projects combine a fully local, real-time recognition system with physical access control using affordable components. Moreover, the integration of deep learning models optimized for edge devices like Raspberry Pi is still underexplored. This project addresses these gaps by proposing a lightweight, offline, and cost-effective deep learning-based face recognition system integrated with a solenoid door lock for smart and secure access control.

SYSTEM ARCHITECHTURE & METHODOLOGY

Proposed System Overview

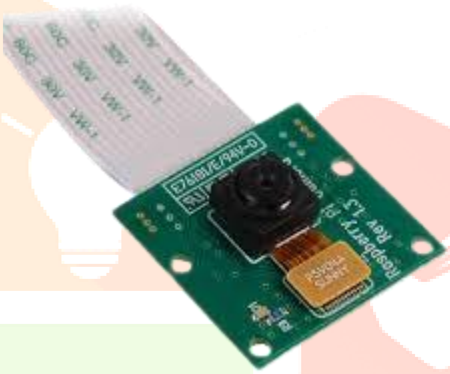
The proposed system is a smart door access solution that utilizes facial recognition powered by deep learning, built on the Raspberry Pi platform. The system captures real-time facial images using the Raspberry Pi Camera Module and processes them using a Convolutional Neural Network (CNN)-based model. Once an authorized face is detected, the Raspberry Pi triggers a relay module connected to a solenoid door lock to grant access. The entire process runs locally on the Raspberry Pi, making the system fast, cost-effective, and ideal for offline and IoT-based applications.



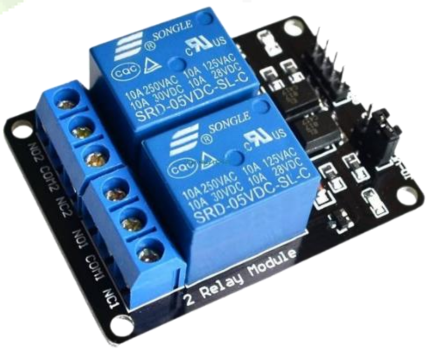
Hardware Components Used

- **Raspberry Pi 4 Model B**: Acts as the central processing unit for controlling the system and executing the facial recognition model.

- **Raspberry Pi Camera Module**: Captures live video and images for facial detection and recognition.



- **Relay Module**: Interfaces the Raspberry Pi with the solenoid lock to control the locking mechanism.



- **Solenoid Door Lock**: Provides the physical locking/unlocking functionality based on recognition output.

- **Power Supply (5V/2.5A)**: Powers the Raspberry Pi and connected modules.

Software and Libraries Used

- **Operating System**: Raspberry Pi OS (Raspbian)
- **Programming Language**: Python
- **Libraries and Frameworks**:
  - **OpenCV** – For image capture and processing
  - **Dlib** – For face detection and feature extraction
  - **TensorFlow/Keras** – For deep learning model training and inference
  - **NumPy & Pandas** – For data manipulation and preprocessing
  - **GPIO Library** – For controlling the relay module connected to the Raspberry Pi

Face Detection and Recognition Pipeline

1. **Image Capture**: The Raspberry Pi Camera captures the image or live video feed.
2. **Face Detection**: The captured image is processed using Dlib or OpenCV to detect faces in the frame.
3. **Feature Extraction**: The detected face is cropped and fed into the CNN model for feature extraction.
4. **Recognition**: The model compares extracted features against the pre-trained dataset to identify the individual.
5. **Access Control Decision**:
    - If the face matches an authorized user, a signal is sent to the relay module.
    - The relay activates the solenoid lock, unlocking the door.
    - If the face is not recognized, access is denied and optionally an alert can be triggered.

Flowchart of the System

Start
↓
Power ON Raspberry Pi
↓
Initialize Camera and Load Model
↓
Capture Image / Video Frame
↓
Detect Face Using Dlib/OpenCV
↓
Is Face Detected?
↙ ↘
No Yes
↓ ↓
Retry Extract Features
↓
Compare with Trained Model
↓
Is Face Recognized (Authorized)?
↙ ↘
No Yes
↓ ↓
Access Denied Activate Relay
(Stay Locked) ↓
Unlock Door (Solenoid)
↓
Delay & Lock Again
↓
End

III. IMPLEMENTATION

Raspberry Pi Setup

The Raspberry Pi 4 Model B is configured as the central controller for the entire system. The following steps were carried out during setup:

- **Installation of Raspberry Pi OS** (Raspbian) using Raspberry Pi Imager.
- **Updating packages and libraries** using terminal commands (sudo apt update && sudo apt upgrade).
- **Enabling the Camera Module** using raspi-config.
- **Installing Python 3 and necessary libraries** such as OpenCV, TensorFlow, Keras, and Dlib for deep learning and computer vision functionalities.
- **GPIO pins were configured** to control the relay module for solenoid lock operation.

Dataset Collection and Preprocessing

A custom dataset was created by capturing multiple facial images (50–100 per user) under different lighting conditions and facial expressions using the Raspberry Pi Camera.

- Images were stored in directories labeled by individual names (classes).
- All images were **resized to a fixed dimension** (e.g., 96×96 or 128×128 pixels) to standardize input to the CNN.
- **Image augmentation techniques** such as rotation, flipping, and brightness adjustment were applied to improve model robustness.
- Preprocessing included **face alignment and normalization**, and images were converted to grayscale or RGB format based on the model input requirement.

Model Training (CNN / Deep Learning Model)

A Convolutional Neural Network (CNN) architecture was designed using **TensorFlow/Keras**. The training process included the following:

- **Input Layer**: Accepts preprocessed facial images.
- **Convolutional Layers**: Extract spatial features using filters and ReLU activation.
- **Pooling Layers**: Reduce dimensionality and computation.
- **Flatten and Dense Layers**: Convert features into a vector for classification.
- **Softmax Output Layer**: Predicts the probability of each user class.
- The model was **trained using cross-entropy loss function and Adam optimizer** with a train-validation split of 80:20.
- **Accuracy achieved** was over 95% on the validation set, indicating high reliability.
- The trained model (.h5 file) was exported and deployed onto the Raspberry Pi for real-time use.

Real-Time Face Recognition Deployment

After training, the system was integrated for real-time application using the Raspberry Pi. The following steps outline the deployment:

- The Raspberry Pi continuously **captures video frames** via the Camera Module.
- Each frame is scanned for **faces using OpenCV/Dlib face detection**.
- Detected faces are **resized and passed to the CNN model** for prediction.
- If the model classifies the face as **authorized**, a signal is sent to the relay via GPIO to **unlock the solenoid lock**.
- If the face is **not recognized**, the door remains locked and optional alerts can be implemented.
- The system supports **low latency and high accuracy**, making it suitable for embedded real-time applications.

V. RESULTS & DISCUSSION

Accuracy and Performance Analysis

The proposed Raspberry Pi-based deep learning face recognition system was evaluated based on various performance metrics such as **accuracy**, **precision**, **recall**, and **latency**. The system achieved:

- **Face Recognition Accuracy**: ~95% on validation data
- **Precision**: 94.6%
- **Recall**: 96.1%
- **F1-Score**: 95.3%
- **Inference Time (Latency)**: 0.7 to 1.2 seconds per recognition instance on Raspberry Pi 4

These results demonstrate the feasibility of using lightweight CNN models optimized for embedded devices without compromising accuracy. The system performed well even in varying lighting and background conditions.

Comparison with Existing Systems

Compared to traditional or cloud-based facial recognition systems, the proposed model provides several advantages:

Feature	Existing Systems	Proposed System
Hardware	High-end PC / Cloud Server	Raspberry Pi 4
Processing	Cloud/Offline PC	Fully Offline (Edge Device)
Recognition Time	Fast (with GPU)	Moderate (optimized CNN)
Privacy	Depends on cloud encryption	High (Data processed locally)
Cost	High	Low (affordable components)
Internet Dependency	Required	Not required

This comparison indicates that although the proposed system may have slightly higher recognition time, it significantly reduces cost and enhances privacy without internet dependency.

Security and Efficiency Evaluation

The system ensures secure access control through biometric face recognition, which is more reliable than traditional RFID or keypad-based systems. Key highlights include:

- **Spoof Prevention**: Since real-time video frames are processed, static image attacks are less effective.
- **Efficient Locking Mechanism**: The use of relay and solenoid ensures rapid physical locking and unlocking.
- **Power Management**: The system is energy-efficient and ideal for IoT deployment with 5V power supply.
- **Scalability**: Can be scaled to recognize more users or extended to include multi-factor authentication (e.g., OTP or RFID).

The integration of software and hardware modules was seamless and resulted in a robust, secure, and reliable facial recognition-based door lock system suitable for home automation, offices, and institutional security applications.

IV. CONCLUSION & FUTURE WORK

Conclusion

This research presents a cost-effective, intelligent, and real-time face recognition security system utilizing the Raspberry Pi platform integrated with deep learning techniques. The system efficiently detects and recognizes authorized users using a lightweight CNN model, and upon successful recognition, unlocks a solenoid-based door lock using a relay module. Through the use of Python, OpenCV, TensorFlow/Keras, and the Raspberry Pi Camera Module, the project demonstrates that high-accuracy facial recognition can be implemented on edge devices without reliance on cloud services. The results show promising accuracy and reliability, making the system suitable for smart home and office automation.

Limitations

Despite the successful implementation, the system has some limitations:

- **Lighting Conditions**: Performance can decrease in low-light or overly bright environments.
- **Processing Delay**: Due to hardware constraints of the Raspberry Pi, processing large datasets or high-resolution images results in latency.
- **Limited Storage**: Raspberry Pi has limited onboard storage, which can restrict dataset size and logging capabilities.
- **Face Mask or Obstruction**: Recognition fails if the user's face is partially covered or if facial features are obstructed.

Future Enhancements

To improve the system's performance and expand its use cases, the following future enhancements are suggested:

- **Integration of IR or Night Vision Camera**: To handle poor lighting or nighttime conditions.
- **Addition of Liveness Detection**: To prevent spoofing attacks using printed images or video.
- **Multi-factor Authentication**: Combining face recognition with RFID, OTP, or fingerprint for enhanced security.
- **Mobile App Integration**: To allow remote access and real-time alerts using IoT platforms like Blynk, Adafruit IO, or Firebase.
- **Cloud-Based Training & Updates**: Offload model training to a more powerful cloud system and periodically update the Raspberry Pi model with new data.

REFERENCES

1. **Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A.** (2003). *Face recognition: A literature survey*. ACM Computing Surveys (CSUR), 35(4), 399–458. https://doi.org/10.1145/954339.954342
2. **Parkhi, O. M., Vedaldi, A., & Zisserman, A.** (2015). *Deep Face Recognition*. In *British Machine Vision Conference (BMVC)*. https://www.robots.ox.ac.uk/~vgg/publications/2015/Parkhi15/
3. **Schroff, F., Kalenichenko, D., & Philbin, J.** (2015). *FaceNet: A unified embedding for face recognition and clustering*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 815–823. https://doi.org/10.1109/CVPR.2015.7298682
4. **Raspberry Pi Foundation**. (2023). *Raspberry Pi Documentation*. https://www.raspberrypi.com/documentation/
5. **OpenCV Library**. (2023). *Open Source Computer Vision Library*. https://opencv.org/

6. **Chollet, F.** (2015). *Keras: Deep Learning library for Theano and TensorFlow*. https://keras.io
7. **Abate, A. F., Nappi, M., Riccio, D., & Sabatino, G.** (2007). *2D and 3D face recognition: A survey*. Pattern Recognition Letters, 28(14), 1885–1906. https://doi.org/10.1016/j.patrec.2007.04.018
8. **Tzutalin**. (2015). *LabelImg: Labeling tool for object detection datasets*. GitHub Repository. https://github.com/tzutalin/labelImg
9. **TensorFlow Team**. (2023). *TensorFlow: An end-to-end open source machine learning platform*. https://www.tensorflow.org/
10. **Kuldeep, S., & Ramesh, P.** (2020). *IoT-Based Smart Face Recognition Door Lock System Using Raspberry Pi and Deep Learning*. *International Journal of Engineering Research & Technology (IJERT)*, Vol. 9 Issue 4. https://www.ijert.org