Face Recognition using Deep Learning

Banumalar *Koodalsamy[1]*,Manikandan Bairavan Veerayan[1], and Vanaja Narayanasamy[1]*
[1] Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India

Abstract. Identifying a person primarily relies on their facial features, which even distinguish identical twins. As a result, facial recognition and identification become crucial for distinguishing individuals. Biometric authentication technology, specifically facial recognition systems, are utilized to verify one's identity. This technology has gained popularity in modern applications, such as phone unlock systems, criminal identification systems, and home security systems. Due to its reliance on a facial image rather than external factors like a card or key, this method is considered more secure. The process of recognizing a person involves two primary steps: face detection and face identification. This article delves into the concept of developing a face recognition system utilizing Python's OpenCV library through deep learning. Due to its exceptional accuracy, deep learning is an ideal method for facial recognition. The proposed approach involves utilizing the Haar cascade techniques for face detection, followed by the following steps for face identification. To begin with, facial features are extracted through a combination of CNN methods and the linear binary pattern histogram (LBPH) algorithm. For attendance to be marked as "present," the check-in and check-out times of the detected face must be legitimate. If not, the face will be displayed as "unknown."

1. INTRODUCTION

The process of recognizing a person by studying their facial traits is known as face recognition. By enabling extremely precise and effective recognition algorithms, deep learning has revolutionized the field of face recognition. A huge collection of facial photos is often used to train a neural network for deep learning face recognition systems in order to discover the underlying patterns and features that distinguish one face from another. The representation of each person's face that is produced using these features is then individually saved in a database for eventual recognition. The system determines which match is the closest by comparing the input face image with the stored representations during recognition. To extract features from images for face recognition systems, convolutional neural networks (CNNs) are frequently utilized, and deep learning-based methods like Siamese networks and triplet networks are used to optimize the matching process. Face recognition technology has several uses, including social media tagging,

*Corresponding author: kbanumalar@mepcoeng.ac.in

access control, security and surveillance, and personal identification. However, during the design and implementation of these systems, privacy issues and potential biases must be carefully taken into account and addressed.

The methodological approach that separates and recognizes a face from the continuous stream that tracks, contrasts, and stores information about known people is discussed. A suitable answer is an automatic facial detection classifier. These techniques typically locate a collection of base photos and depict faces as a linear combination of those images. Among these techniques, principal component analysis (PCA) is a well-known illustration. PCA simply uses pairwise correlations between pixels in the picture database to determine the basis images [1]. In contrast to Principal Component Analysis (PCA), which is frequently used in computerized face recognition and authentication tasks, Independent Component Analysis (ICA). The Haar feature-based cascade classifier is used for face detection and face localization in face recognition systems. Utilizing the weighted Local Binary Pattern technique, facial features are retrieved. Tested on the FERET database, the suggested embedded facial recognition system achieves accuracy of CMC: 99.33% and EER: 1% [2].

In order to gain access to high security systems and facilities, this article intends to advance face recognition technology to the point where it can take the place of RF I-Cards and passwords.The human face is a key factor in identifying a person. Even identical twins have distinctive faces. In order to distinguish one another, facial recognition and identification are necessary. The verification technology to establish a person's identity using biometrics is a facial recognition system.

The most crucial feature in identifying anybody is their face. Face recognition aids in the authentication of every person's identity utilising his unique personal traits because it serves as a distinct identity for everyone. Face Recognition is the skill of identifying and recognising someone based on their face features. Face is multidimensional, thus many mathematical calculations are necessary [3]. A common issue with artificial intelligence is face recognition. Our daily lives made significant use of this programme. Face recognition has been approached in a number of ways up to this point, but it is still exceedingly challenging in practical settings [4].This research presented a strong and straightforward technique for face identification based on camera-captured images called PCA and Back Propagation Neural Network. Face detection, face feature extraction, and face recognition are the three stages of the proposed approach [5].

Deep learning techniques are effective but frequently necessitate costly computations and produce complex models that need a lot of data to be trained. A face recognition system based on a recent technique that uses artificial neural networks to address both face representation and recognition is described [6]. In order to offer a measurement for automatic face identification, this study shows how to transform the front of a human face to image coding and a histogram of pixel location [7]. Convolutional Neural Network (CNN)-based real-time facial recognition technology is designed and evaluated in this research. Standard AT&T datasets are used for the initial evaluation of the proposed design, and the results are then expanded to the creation of a real-time system [8-9]. Face recognition technology, which exclusively uses faces for attendance, can solve the first issue. But choosing the right algorithm to use is challenging due to the abundance of facial recognition algorithms [10]. In computer vision and deep learning, face detection and recognition are developing and active research fields. There are several uses for face detection and identification, including identifying persons in certain settings like supermarkets and banks.

2. System architecture and design

The attendance for this work is captured through facial recognition technology, with the students' information being stored and updated on a website built with MY SQL and PHP as shown in Figure 1. The Raspberry Pi camera is used to take photos of the students' faces. To ensure the project's success as an attendance system, it's necessary to establish a connection to a wifi network and have USB ports available for a monitor, keyboard, and mouse as shown in the figure1. A camera module is employed to authenticate users by capturing their face and comparing it to the database images for access. A website was developed to manage student attendance online, which records information about each student.To access and confirm their attendance on the website, a student must provide specific information such as their student name, register number, department, year, semester, email address, and a secure password. They may use their register number as their username and the previously established password. The system captures and stores a single student's face from 300 different angles in the database, along with all the relevant information.

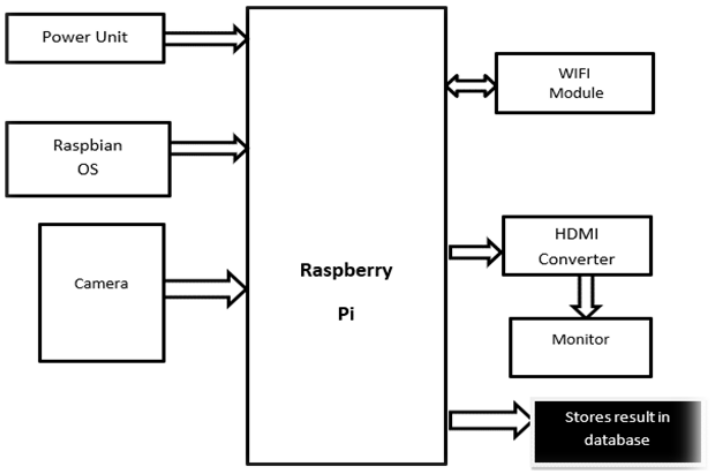

Fig 1. Block Diagram of proposed system

3. Hardware components

Figure 2 illustrates the controlled module of the designed system that employs Raspberry Pi model 3 B. The Pi 4 is equipped with a 64-bit ARM Cortex A72 and 4GB of RAM, and also features a Video Core VI graphical processing unit (GPU) for graphical processing applications (GPA). In addition, the Pi 4 includes two USB ports and 40 GPIO pins for connecting external electronic devices, which were utilized by the door locking/unlocking module through the GPIO pins. Raspberry Pi is specifically designed to run a Linux-based operating system (LOS) and has its own operating system, Raspbian (ROS), which utilizes Python as its official programming language.

Fig 2. Raspberry Pi model 3 B

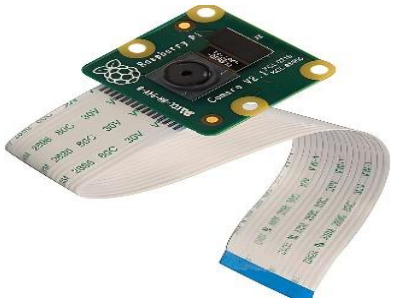

Fig 3. Raspberry Pi camera module

Figure 3 displays the Raspberry Pi Camera Module 3, which is a small-sized camera produced by Raspberry Pi. It boasts an IMX708 12-megapixel sensor with HDR and phase detection autofocus. The Camera Module 3 comes in standard and wide-angle versions, each of which can be obtained with or without an infrared cut filter. This camera can capture full HD video and still photographs, with an HDR mode that supports up to 3 megapixels. It is fully supported by the libscamera library, which includes its rapid autofocus feature, making it user-friendly for novices while providing ample capabilities for advanced users. Additionally, Camera Module 3 is compatible with all Raspberry Pi computers. Figure4 displays the overall hardware Setup of this work.

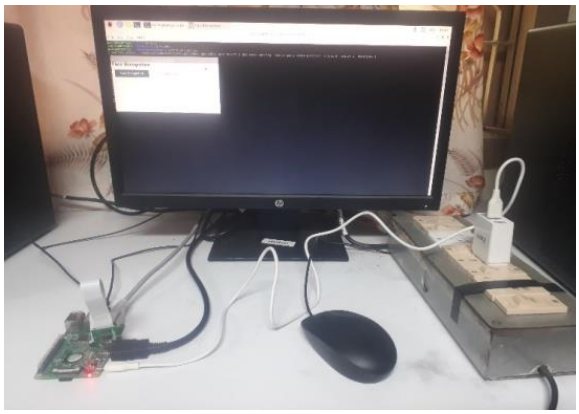

Fig 4. Hardware Setup

4. System description

The Raspberry Pi camera connects to the Raspberry Pi through its USB ports. Raspberry Pi is designed to run a Linux-based operating system (LOS) and has its own operating system

called Raspbian (ROS), which employs Python as its official programming language. The camera detects and identifies whether a person is authorized or not. The control system is responsible for relaying information regarding presence or absence, which is achieved using Python programming code. Flow diagram as shown in figure 5.

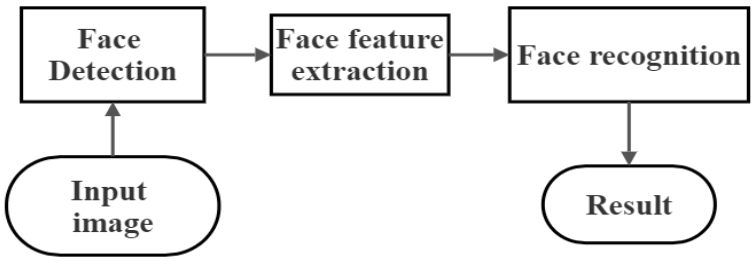

Fig 5. Flow diagram of proposed system

4.1 Face image dataset:

This repository contains a collection of 9,330 images of 30 students' faces as shown in Figure6. The images were captured using a camera and automatically cropped using the OpenCV face library. To improve recognition, the images were taken from different angles. To simplify processing and avoid complexity in the model, the detected color images were converted to grayscale.

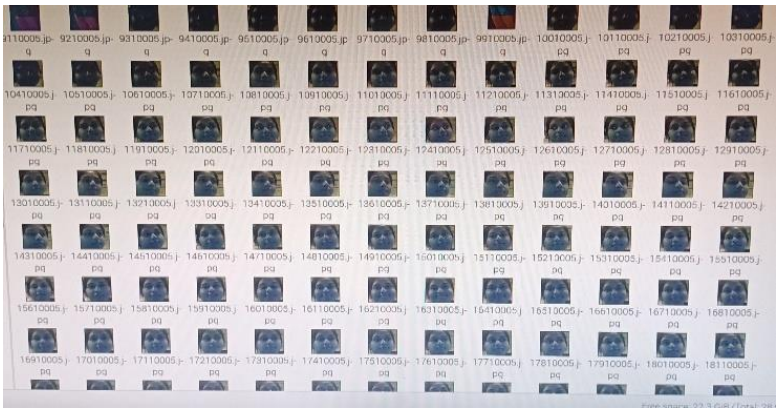

Fig 6. Dataset Collection

4.2 Pretrained CNN model:

CNNs, or convolutional neural networks, are neural networks commonly employed in tasks involving image and video recognition. They are specifically designed to learn spatial hierarchies of features from raw input data in an automatic and adaptive manner. Figure 7 represents about CNN model which is described below.

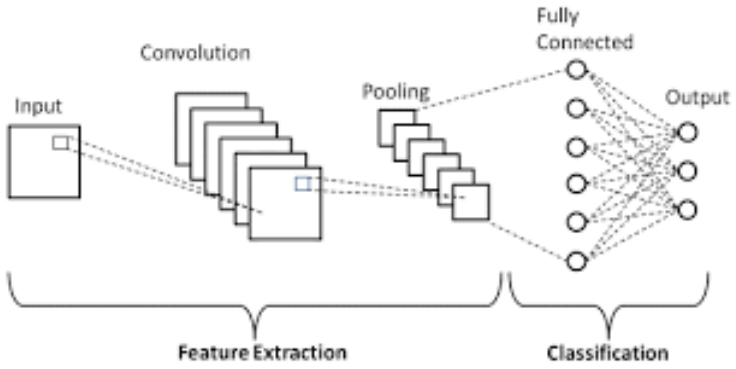

Fig 7. CNN Model

4.3 Feature extraction:

- **INPUT:**Initially, the input image of a person is provided.
- **CONVOLUTION:**Convolution is a mathematical process that enables the combination of two sets of information. In face comparison tasks using CNNs, a pair of faces is inputted independently for feature extraction. Both faces are subjected to the same filters, ensuring that the representation of a face remains constant regardless of the other face it is being compared to.
- **POOLING:** Pooling is the step in which features are extracted from the image output of a convolutional layer. This involves reducing the dimensionality of the image by extracting its key features, and combining the resulting output of the previous layer into a single one.

4.4 Classification:

- **FULLY CONNECTED LAYER:** After the pooling process, a fully connected layer is employed to predict the most suitable label for the given image. This involves flattening the output from the previous layers and feeding it into the FC layer. The flattened vector then undergoes additional FC layers where mathematical operations are performed. This stage marks the beginning of the classification process. Two fully connected layers are connected because it has been observed that this configuration performs better than using a single connected layer.

  OUTPUT: Finally the classified and extracted images are given as output as shown in the Figure 8. In this case, the ResNet50 CNN, depicted in Figure 8, is utilized as a pre-trained model for extracting facial features and performing classification.

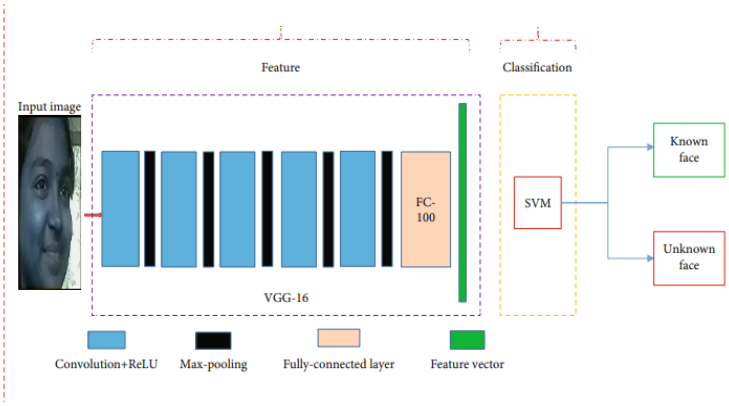

Fig 8. Extraction of images

4.5 Resnet-50 with SVM model

ResNet-50 is a deep convolutional neural network that can be used for feature extraction in face recognition systems. SVM is a machine learning algorithm that can be used for classification based on the extracted features. When combined, ResNet-50 with SVM can improve the accuracy of face recognition systems. Here is how ResNet-50 with SVM works on face recognition:Face Detection: First, the input image is passed through a Haar Cascade classifier or other face detection algorithms to identify the face region.

1. Feature Extraction: The face region is then passed through a ResNet-50 deep neural network, which extracts high-level features that are robust to variations in pose, lighting, and other factors. ResNet-50 is a pre-trained neural network that has been trained on a large dataset of faces, allowing it to learn powerful representations of facial features.
2. Feature Encoding: The extracted features are then encoded into a fixed-length feature vector that can be used for comparison and classification.
3. SVM Classification: The encoded features are passed through an SVM classifier, which learns to classify the face based on the extracted features. The SVM model can be trained using a set of labeled face images to learn to distinguish between different faces.
4. Face Recognition: During the recognition phase, a new face image is compared to the labeled face images in the dataset by passing the new image through the same pipeline. The face is first detected, then passed through the ResNet-50 for feature extraction, and finally through the SVM classifier for classification. The system then returns the identity of the closest match in the dataset based on the output of the SVM.

4.6 Support vector machines:

SVM (Support Vector Machines) is a machine learning algorithm that can be used for classification and regression analysis. In face recognition, SVM can be used to classify faces based on the extracted features. In face recognition, the input image is first passed through a face detection algorithm to detect and extract the face region. Next, features are extracted from the face region using the technique Local Binary Patterns (LBP). These features are then used to represent the face in a feature space, where they can be compared to other faces to perform recognition.SVM can be used to classify the face based on the

extracted features. The SVM algorithm learns to separate the different classes of faces in the feature space using a hyperplane that maximizes the margin between the classes. The hyperplane is learned during the training phase of the SVM, where a set of labeled face images is used to train the classifier to distinguish between different faces. During the recognition phase, a new face image is compared to the labeled face images in the dataset by passing the new image through the same feature extraction pipeline. The features are then passed through the SVM classifier, which outputs the class label of the closest match in the dataset based on the distance between the feature vectors.

5 Train the model

5.1 Haarcascade classifier

Haar cascade classifiers are a type of object detection algorithm that can be used to detect objects in images or video, including faces.AHaar cascade classifier works by analyzing an image at different scales and sizes to identify features that are characteristic of the object being detected. In the case of face recognition, the algorithm looks for features like the eyes, nose, and mouth.The process begins with creating a "cascade" of classifiers, which are trained on positive and negative examples of the object being detected.

In the case of face detection, positive examples would be images of faces, while negative examples would be images without faces. The algorithm learns to identify the features that are most likely to be present in a face, while filtering out features that are not relevant.Once the cascade has been trained, it is applied to an image by scanning the image with a sliding window at different scales and sizes. At each location, the algorithm checks whether the features being detected are present. If enough features are present, the algorithm concludes that a face has been detected at that location.

Here's a explanation of how a Haar cascade classifier works for face detection:

1. Training Stage: During the training stage, the algorithm is trained on a large dataset of images containing faces (positive samples) and images not containing faces (negative samples).
2. Feature Extraction: The algorithm extracts features from each image in the dataset. Haar-like features are used to capture the differences in intensity between adjacent regions of an image.
3. Adaboost Learning: Adaboost is a machine learning algorithm that is used to combine multiple weak classifiers into a strong classifier. Each weak classifier is trained to detect a specific feature in the image, and Adaboost selects the best features to create the strong classifier.
4. Cascade of Classifiers: The final strong classifier is a cascade of multiple classifiers. Each classifier in the cascade is trained to detect a different set of features in the image. The cascade works by rejecting regions of the image that are unlikely to contain the object being detected, while passing promising regions on to the next classifier in the cascade.
5. Detection: During the detection stage, the cascade is applied to a new image. The algorithm scans the image at different scales and locations, and at each location, the features of the image are compared to the features in the cascade. If the features match those of a face, then the algorithm identifies the region as a face.
   Figure 9 illustrates how Haar-like features are used to extract facial features, such as intensity and length, from an image.

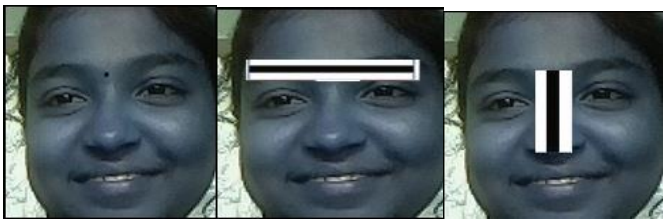

Fig 9. Haar Features of Face

5.2 Local binary pattern histogram

LBPH is an algorithm that is employed for face recognition purposes. Figure 10 dispalys the work flow of LBP Algorithm. The algorithm involves five steps, which are outlined below: LBP Parameters: The LBPH algorithm relies on four key parameters, namely, the radius, neighbors, grid X, and grid Y. The radius is utilized to construct a circular local binary pattern and refers to the distance around the central pixel. Neighbors denote the number of sample points utilized in creating the circular local binary pattern. Grid X and Grid Y represent the number of cells in the horizontal and vertical directions, respectively.

1. Radius: The algorithm calculates the radius of the face image by measuring the distance between the eyes, nose, jaws, lips, and forehead.
2. Radius: The algorithm calculates the radius of the face image by measuring the distance between the eyes, nose, jaws, lips, and forehead.
3. Neighbors: A graph is created by connecting the central points of facial features such as eyes, nose, lips, and cheeks.
4. Grid X: The width of the face is measured horizontally.
5. Grid Y: The height of the face is measured vertically.

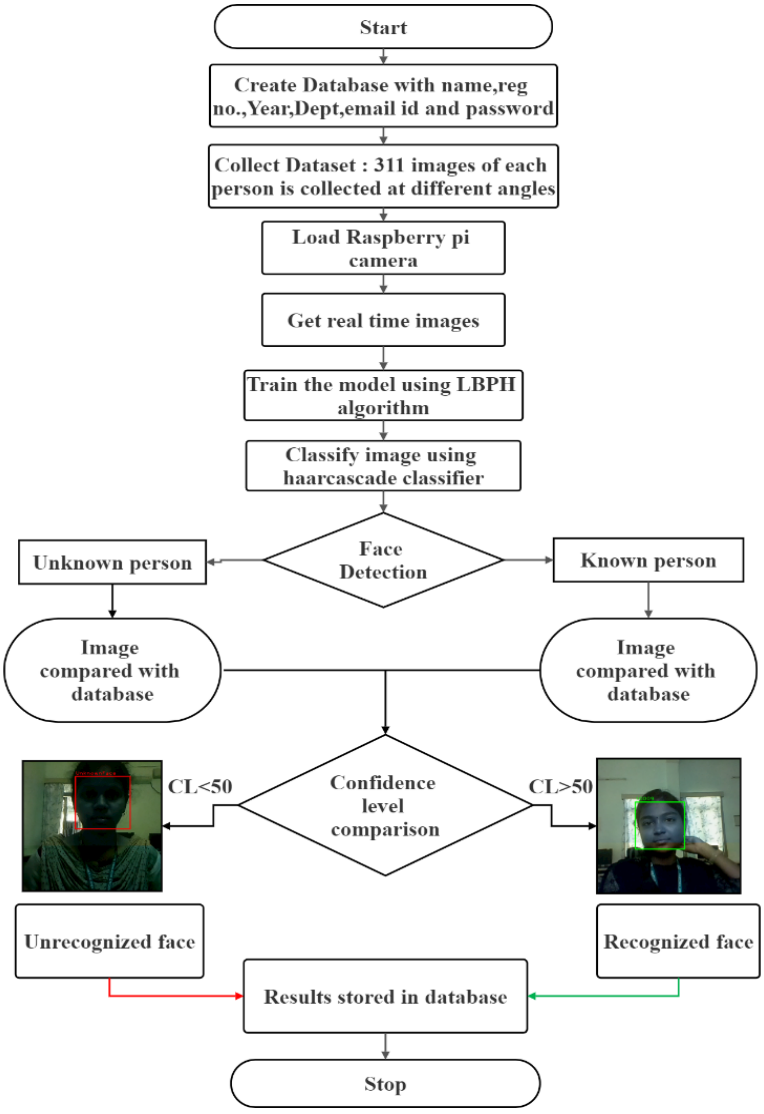

Fig 10. Flowchart of LBPH Algorithm

The LBPH algorithm operates in the following manner:

1. Radius: The algorithm calculates the radius of the face image by measuring the distance between the nose, eyes, jaws, lips, and forehead, which is subsequently noted.
2. Neighbors: A graph is generated by connecting the central points of the eyes, nose, lips, and cheeks on the face.
3. Grid X: The face's width is measured horizontally as shown in the figure 11.
4. Grid Y: The face's height is measured vertically as shown in the figure 11.

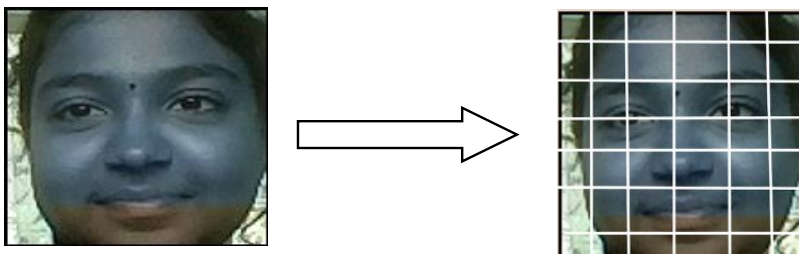

Fig 11. Representation of Grid X and Grid Y

5. Training the Algorithm: The first step is to train the algorithm using a dataset that contains images of the individuals we wish to recognize. Each image in the dataset must be assigned a unique identifier, which can be a number or name of the person. The algorithm utilizes this information to recognize an input image and produce an output. Images of the same person should have the same identifier. Once the training dataset is prepared, we can observe the computational steps involved in LBPH.

6. Applying the LBP operation: the initial computational stage involves applying the LBP operation to generate an intermediary image that accentuates the facial features and better represents the original image. This is achieved through the implementation of a sliding window approach that takes into account the radius and neighbors parameters.

7. Extracting the Histograms: With the image produced in the previous step, we can partition it into several grids by employing the Grid X and Grid Y parameters.

8. Performing the face recognition: At this stage, the algorithm has undergone training. The resulting histograms are utilized to represent each image from the training dataset.

5.3 LBPH Workflow

- A camera module for capturing real-time images
- The ability to capture and store a photograph and then compare it to an image in the database
- If the confidence level (CL) assigned to the facial recognition process is equal to or greater than 50, the student is deemed present. Conversely, if the CL is below 50, the student is marked as absent.
- The present and absent status is logged in the SQL database, which has been created and integrated into the website, as depicted in Figure 11.

5.4 Face recognition

The final stage of the process is face recognition. The camera captures real-time images and compares them to the dataset after the registered id of the individual has been provided. If the captured image corresponds to an image in the dataset, the person is marked as present, otherwise, they are marked as absent. The present and absent status can be viewed through the website.

6. Results

Our model has achieved success in detecting and recognizing the faces of individuals in real-time video frames. The Python code has been efficiently programmed and tested with Raspberry Pi model 3 B. The facial recognition process involves combining CNN with Local Binary Pattern Histogram (LBPH) to extract the relevant facial features. The attendance status of students is updated and marked as present through the website.

Conclusion

Face recognition using deep learning is a powerful technology that has made great strides in recent years. Deep learning models can identify and recognize faces with high accuracy, even in challenging scenarios such as low light, occlusions, and pose variations.To build a

face recognition system using deep learning, the first step is to gather a large dataset of face images and use it to train a deep neural network. This network can then be used to extract facial features from new images and compare them to those in the training set to recognize individuals. The combination of resnet, SVM and haar cascade techniques can improve the accuracy of face recognition systems. First, the Haar Cascade classifier can be used to identify the face region in the image, reducing the search space for subsequent processing steps. Then, the face region can be passed through a ResNet to extract features that are robust to variations in pose, lighting, and other factors. Finally, SVM can be used to classify the face based on the extracted features. Overall, face recognition using deep learning has tremendous potential for a wide range of applications, from security and surveillance to personalization and entertainment. As the technology continues to evolve, we can expect to see even more sophisticated and accurate face recognition systems in the future.

References

1. Kennedy O kokpujie, Etinosa Noma-Osaghae, Olatunji J. Okesola, Samuel N. John, Okonigene Robert, International Conf on Computational Science and Computational Intelligence (2017).
2. Paul viola and Michael J. international J of computer vision, 57, **2**, 137-154 (2004).
3. Rajesh Kumar Misra, Satyanrayan Padhy, Sandipan Pine, Prabhat Kumar Patnaik, N.Jeevaratnam, Indian J of Natural Sciences, 13 , 72 (2022).
4. Marian Stewart Bartlett, Member, Javier R. Movellan, Member, Terrence J. Sejnowsk, IEEE transactions on neural networks, 13, 6, (2002).
5. C. Havran, L. Hupet, J. Czyz, J. Lee, L. Vandendorpe, M. Verleysen, IEEE Trans on Neural Networks, 13, 6, (2002).
6. Shahrin Azuan Nazeer, Nazaruddin Omar, Marzuki Khalid, International J of Security and its Applications 10, 3, 81-100 (2016).
7. Thai Hoang Le, Advance in Artificial Neural System, 17 (2017).
8. K. B. Pranav, J.Manikandan, Third International Conf on Computing and Network Communications 25, (2018).
9. Neel Ramakant Borkar; Sonia Kuwelkar, International Conf on Computing Methodologies and Communication (ICCMC), (2018)
10. Zaid Alyasseri in International J of Computer Applications 126(3), 34-38(2015).