Real Time Face Recognition System using Raspberry Pi4

BOUKERCHE Alaa Eddine
Faculty of Electrical Engineering, Telecommunications Department. University of Science and Technology Houari Boumediene, USTHB.
alaaboukerche08@gmail.com

CHELALI Fatma Zohra
Speech Communication and Signal Processing Laboratory, Faculty of Electrical Engineering, University of Science and Technology Houari Boumediene, USTHB.
chelali_zohra@yahoo.fr fchelali@usthb.dz

Abstract— **Modern technologies like computer vision (CV), Artificial Intelligence (AI), the internet of things (IoT), and (AIoT) Artificial Intelligence of Things have advanced in many fields, particularly surveillance systems, which need to process faces in real time to guarantee security. This paper proposed a comparative analysis for a real time face recognition system from video sequences with two architectures including traditional application using a personal computer and a Raspberry Pi, respectively. Our study offers an implementation for a real-time face recognition system employing Viola-Jones algorithm and Deep neural networks techniques such as Convolutional Neural Networks (CNN) for face detection and face recognition. Our face recognition system seeks to replace the conventional approach of keeping track of attendance by implementing a Raspberry Pi-based attendance monitoring system that can be utilized by all institutions and organizations. Significant progress has been made thanks to the use of CNN models which has allowed an accuracy rate of 94%.**

Keywords—Real time face recognition; Raspberry Pi; CNN; Open CV; Python.

I. INTRODUCTION

Facial recognition is a method of authenticating or verifying the identity of an individual by analyzing their facial features. Over the years, this technology has gained popularity and its use has spread in many fields, offering diverse and varied applications.

Facial recognition has made great progress in recent years. It is used in several fields of application such as securing access such as banks, biometric systems, human machine interfaces, multimedia applications, speech and speaker recognition, computer vision applications, face synthesis, etc.

The computer vision research community is receptive to research in the area of image and video classification [1]. Applications for image and video classification include face recognition, industrial automation, security monitoring, medical image analysis, content-based multimedia analysis, and remote sensing.

Several techniques are applied to recognize an individual from his/her face, we mainly find global methods, local methods and hybrid methods. The results obtained are discussed according to the efficiency obtained and the execution time necessary for the learning and testing phase.

The recognition of individuals in real time is of great interest due to the multitude of applications which require the verification of the person for access to a company, a service, a hospital, a bank, etc. We find recent methods that use Deep learning, requiring a large database for learning, and other methods requiring machine learning techniques.

In video surveillance, human-computer interface, and application personalization, human face recognition is crucial. In this research, we offer a method that detects a face and compares it with stored data of known individuals in order to detect and identify a human face from real-time video. The implementation of these methods on electronic circuits such as FPGA cards, raspberries and Arduino circuits has evolved in recent years.

Real-time facial recognition, powered by advances in deep learning and machine learning, has revolutionized the way we interact with technology. Thanks to sophisticated systems capable of processing and analyzing visual data in real time, diverse applications are emerging, ranging from security and surveillance to personalizing the user experience. In this work, we will explore how the use of the Raspberry Pi, a versatile and inexpensive computing platform, integrates seamlessly with these technologies to offer effective and accessible facial recognition solutions. By examining technical challenges, recent advances, and potential applications, we will highlight the revolutionary potential of this convergence between real-time facial recognition and the limited resources offered by the Raspberry Pi.

Our face recognition system aims to implement an attendance tracking system using a Raspberry Pi; that can be used in all institutions and organizations such as universities or different companies to avoid the traditional method for recording attendance. In fact, face recognition is the most intelligent, fastest and efficient technique to activate the attending management system. Our system used the stored video

2024 3rd International Conference on Advanced Electrical Engineering (ICAEE) | 979-8-3503-4935-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICAEE61760.2024.10783281

979-8-3503-4935-1/24/$31.00 ©2024 IEEE

sequence of an individual to capture the frames, which will be used later for recognition.

This paper presents a real time face recognition system using Deep neural networks techniques such as Convolutional Neural networks CNN, as well as an implementation using a Raspberry Pi4. Section II highlights a state of the art of real time face recognition systems. Section III discusses our facial recognition system in real time. Finally, a conclusion summarizing our work in given in section IV.

II. STATE OF THE ART

The majority of the work that is currently available discusses software-level facial detection and recognition. Theoretically, very few research publications have examined the combination of software and hardware. FPGA as well as Raspberry have been the most often used platform for the practical implementation of facial detection and recognition utilizing a combination of hardware and software.

A system with an OLED display, capacitive touch sensor, and Raspberry Pi camera module coupled to a Raspberry Pi Zero is presented by the authors in [2]. Their technique employs the Local Binary Pattern Histogram (LBPH) for facial recognition after the Haar Cascade classifier for face detection in an image. OpenCV is used to implement the LBPH method. It was found that 80% of faces could be detected and that 80% of faces could be correctly recognized.

The authors in [3] propose the use of convolutional neural networks (CNNs) for the development and evaluation of a real-time facial recognition system. The initial assessment of the suggested design is conducted using standard AT&T datasets, which are then expanded to include the design of a real-time system. Details on the CNN parameter adjustment that was carried out to assess and enhance the recommended system's recognition accuracy are also provided. Furthermore, a systematic strategy to parameter adjustment is proposed to enhance the system's performance. With conventional datasets and real-time inputs, the recommended method achieves maximum recognition accuracies of 98.75% and 98.00%, respectively.

The authors in [4] describe a method for tracking faces in real-time video and identifying them by comparing them to stored information about people who are known to the tracker. Their method fully ignores any background effect and can identify an individual in a matter of nanoseconds. It also displays more details about that particular person. This approach may also be used in a number of lighting settings, which allows it to function well in a wide range of environments without experiencing substantial errors. They used a Raspberry Pi. Additionally, this method makes use of the advantages provided by many Python libraries, such as NumPy and OpenCV. Its memory is based on the SQLite database management technology. With its quicker and genuine.

In [5], two facial recognition methods for the Raspberry-Pi smart door system are experimentally compared. The first method uses histograms of local-binary patterns. Deep learning and convolutional networks are used in the second one. The implementations are described in the paper along with the time and accuracy of recognition results. It demonstrates that even with small library sets and restricted Raspberry-Pi resources, the CNN-based technique outperforms LBP in terms of speed and recognition accuracy.

A real-time surveillance system that employs a CNN (Convolutional Neural Network) and a Raspberry Pi for facial recognition is described by Zamir et al. [6]. Before comparing the query image with the dataset based on features and landmark face detection, the system is trained on the labeled dataset to extract various facial characteristics. Lastly, it generates a vote-based result by comparing their looks and votes. The Histogram of Oriented Gradient (HOG), a mid-level feature extractor, and cutting-edge face detection and recognition techniques are used to compare the classification accuracy of the CNN-based system. The maximum accuracy of 98%, 98.24%, 89.39%, and 95.71% was achieved for the VMU, facial recognition, and 14 celebrity datasets, respectively.

The Internet of Things (IoT) is a network of networked components that gathers and exchanges data to assist those devices become more advanced by connecting to software, a network connection, sensors, and other electronic equipment [7]. IoT is widely used in daily life, and security monitoring is a fundamental component of all IoT-based systems [7].

Using popular face detection and recognition techniques as Haar detection and PCA, a Raspberry Pi-based face recognition system was proposed by Gupta's research [8]. For this investigation, they linked a camera, a motor, and an LCD to the Raspberry Pi board. The motor spins to show when the gate is opening and closing, and the LCD shows the name of the individual being identified. They developed a real-time program that utilizes the Raspberry Pi as a gate pass by comparing the scans to data stored in the device.

III. FACE RECOGNITION SYSTEM USING CNN

The system implemented in this paper is a real-time facial recognition application. It starts by acquiring a series of 10 images from a video sequence of each individual's face in different positions and storing them in our database. Each person is associated with a name and a unique ID for easy reference. Using the Viola-Jones algorithm, also known as Haar Cascade, we accurately detect each user's facial areas in the acquired images. Then we preprocess these images by converting them to grayscale and resizing them to a uniform size of 120x120 pixels to simplify further processing using the OpenCV library. Fig 1 shows the diagram of our face recognition system.

A subset of machine learning known as "deep learning" has become incredibly popular recently because of its exceptional capacity to handle and comprehend complex data [9]. Artificial neural networks, or deep learning algorithms, are highly effective in a variety of fields, including computer vision, natural language processing, speech recognition, and reinforcement learning. These algorithms are inspired by the composition and operation of the human brain [10].

Deep learning involves the use of neural networks, which are computational models composed of interconnected layers of nodes, or neurons. These networks are trained on vast amounts of data, adjusting their internal parameters through a process called backpropagation to optimize performance on a specific task [11]. With the ability to automatically learn hierarchical representations from raw data, deep learning algorithms excel in tasks involving large datasets and intricate patterns, often surpassing traditional machine learning techniques.

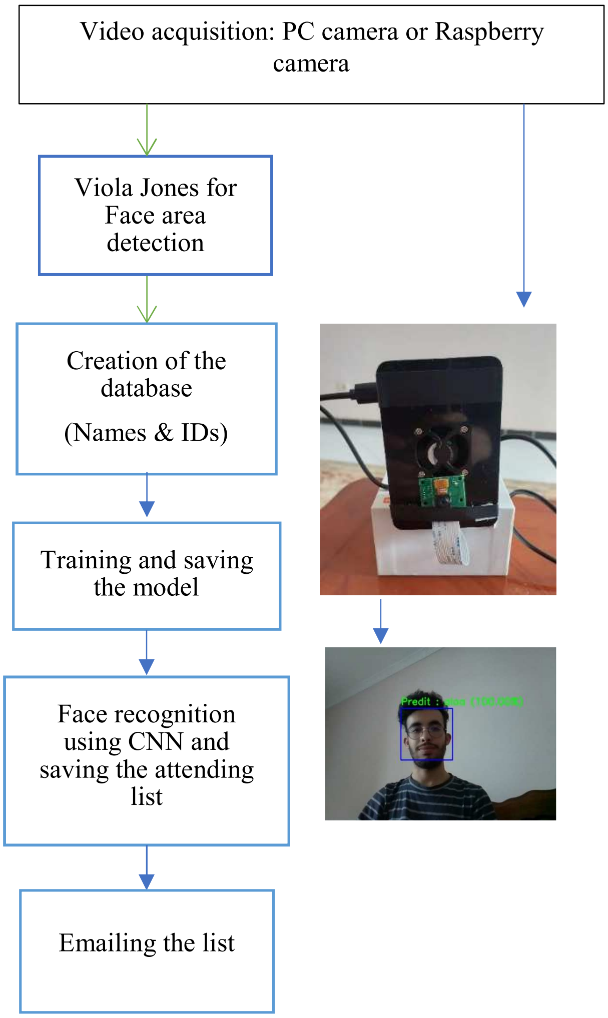

Fig. 1. Diagram of our face recognition system using PC and Raspberry Pi

The Raspberry Pi is among the most widely used IoT devices [7]. The Raspberry Pi is a small, lightweight, and portable computer board that is affordably priced. It can be connected to a keyboard, mouse, television, computer display, and other devices like flash drives.

A. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a specific type of Deep neural network designed for processing structured grid-like data, such as images. CNNs are particularly effective in tasks like image recognition and classification due to their ability to automatically learn hierarchical patterns in data [12,13,14]. They consist of convolutional layers that apply filters to input data, pooling layers that down sample the output of convolutional layers, and fully connected layers that perform classification based on the features learned in the previous layers. CNNs have been instrumental in advancing computer vision applications and have achieved state-of-the-art performance in tasks like object detection, image segmentation, and facial recognition [15,16,1,18,19]. Fig 2 shows a diagram of a basic convolutional neural network (CNN) architecture.

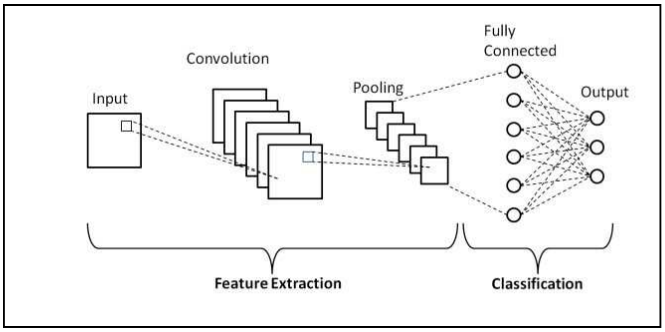

Fig. 2. Schematic diagram of a basic convolutional neural network (CNN) architecture [20]

B. Facial Datasets

we used three datasets for our training phase:

ORL Dataset: The Olivetti Research Laboratory (ORL) dataset contains 400 images of 40 distinct subjects. The images include variations in lighting, facial expressions, and facial details.

FERET Dataset: This dataset is a large collection of facial images that includes 57 subjects with a wide range of variations in terms of age, race, and facial expression, providing a comprehensive set of training data for the model. The total images used is 570.

Face94 Dataset: This dataset contains facial images of 115 individuals taken in a controlled environment with consistent background and lighting, ensuring high-quality images for training and testing the model. It Contains images of male and female subjects with considerable expression changes. Image resolution: 180 by 200 pixels. The total images used is 1150.

C. CNN Architectures and models proposed in our work

We explored three convolutional neural network (CNN) architectures to see the impact of different variations in their complexity, depth and number of parameters, furthermore we trained each model for 100 epochs, which means that the training process was repeated 100 times on the dataset to allow the model to learn and adjust to the training data.

For the first model, the first layer is a convolution layer with 32 filters of size 3x3, followed by a ReLU activation. Then, a second convolution layer with 64 filters of size 3x3 is added, followed by Batch Normalization and ReLU activation. The third layer is a convolution with 64 filters of size 1x1, followed by a dropout layer to prevent overfitting, then Batch Normalization and ReLU activation. Then, a fourth convolution layer with 128 filters of size 3x3 is introduced, followed by a dropout layer and ReLU activation. Finally, a max pooling layer is used to reduce the spatial dimensionality, followed by another convolution with 64 filters of size 1x1 and ReLU activation. After that, the outputs of the previous layers are flattened into a vector and fed into a dense layer of 32 neurons, followed by a SoftMax output layer for classification.

For the second model, the first layer is a convolution layer with 32 filters of size 3x3 and ReLU activation, followed by another convolution layer with 64 filters of size 3x3 and ReLU activation. Then, a max pooling layer is used to reduce the spatial dimensionality. After that, a dropout layer is added to prevent overfitting. Then, three additional convolution layers are added, each followed by a ReLU activation and a max pooling layer to gradually reduce the size of the input image. Finally, the outputs of the convolution layers are flattened into a vector and fed into a dense layer of 32 neurons, followed by a SoftMax output layer for classification.

The third convolutional neural network (CNN) model starts with a first convolution layer using 16 filters of size 3x3. This layer is followed by ReLU activation to introduce non-linearity into the model. Then, a pooling layer is added to reduce the dimensionality of the output. After that, a batch normalization layer is used to stabilize the learning. To avoid overfitting, a dropout layer is inserted. Then, a second convolution layer is added with 32 filters of size 3x3, followed by the same steps of pooling, batch normalization and dropout. The extracted features are then flattened and passed through fully connected layers with ReLU activations and dropout layers for regularization. The final output is produced by a dense layer with SoftMax activation for classification. Table I summarizes the architecture of the three models used in our work.

TABLE I. THE PROPOSED 3 MODELS

CNN Models	**Layers' architecture**
Model 1	Conv (32 filters 3x3, ReLU), Conv (64 filters 3x3, ReLU, BN), Conv(64 filters 1x1, ReLU, Dropout, BN), Conv(128 filters 3x3, ReLU, Dropout), Max Pooling, Conv(64 filters 1x1, ReLU), Dense(32 neurons, ReLU), Dense(Output, SoftMax)
Model 2	Conv (32 filters 3x3, ReLU), Conv(64 filters 3x3, ReLU), Max Pooling, Dropout, Conv(supplementary layers), Dense(32 neurons, ReLU), Dense(Output, SoftMax)
Model 3	Conv (16 filters 3x3, ReLU), Pooling, Batch Normalization, Dropout, Conv (32 filters 3x3, ReLU), Dense(ReLU), Dropout, Dense (Output, SoftMax)

D. Simulation results obtained

Our three models were evaluated and tested using three datasets . For all subjects, images were taken at different times, varying lighting, facial expressions (eyes open/closed, smiling/not smiling), and facial details (glasses/no glasses). 80% of the images were used for training and the remaining 20% for testing.

The training results of the three CNN models were summarized in Table II, containing key metrics such as accuracy, loss, and training time. These metrics provide a comprehensive evaluation of the performance of CNN models, enabling comparison of their effectiveness and robustness.

TABLE II. SIMULATION RESULTS USING THE 3 MODELS

	Accuracy (%)	loss	Training Time (s)
Model 1	88.75	0.8037	5700
Model 2	78.83	0,9379	1020
Model 3	94	0,21	720

After careful analysis of the data presented in the table, Model 3 was identified as the optimal choice to meet the project requirements with low processing time. Its architecture represented in the figure below.

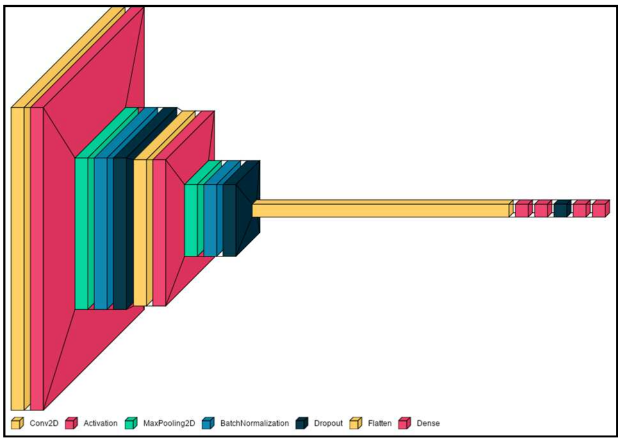

Fig. 3. The architecture of the third model

In addition, the confusion matrix of the third model provides a detailed view of the classification performance as described in figure 4.

Performance metrics such as accuracy, precision, recall, and F1-score are essential for evaluating a classification model's ability to correctly predict sample labels.

Accuracy: Measures the proportion of correctly classified cases compared to the total number of objects in the dataset. It provides an overall effectiveness of the model.

$$\text{Accuracy} = \frac{TP}{Total\ Number\ of\ Predictions} \qquad eq1$$

Where TP stands for True Positives and TN stands for True Negatives.

Precision: Measures the number of correct predictions among all the predictions made by the model. It indicates the exactness of the model.

$$\text{Précision} = \frac{TP}{TP+FP} \qquad \text{eq2}$$

Where FP stands for False Positives.

Recall: Measures the model's ability to identify all positive cases. It captures the model's sensitivity to true positives.

$$\text{Rappel} = \frac{TP}{TP+FN} \qquad \text{eq3}$$

Where FN stands for False Negatives.

F1-Score: A measure of the model's overall accuracy that combines precision and recall.

$$\text{F1 score} = 2 \cdot \frac{\text{Rappel} \cdot \text{Précision}}{\text{Rappel} + \text{Précision}} \qquad \text{eq4}$$

Fig 4 shows the confusion matrix of the third model.

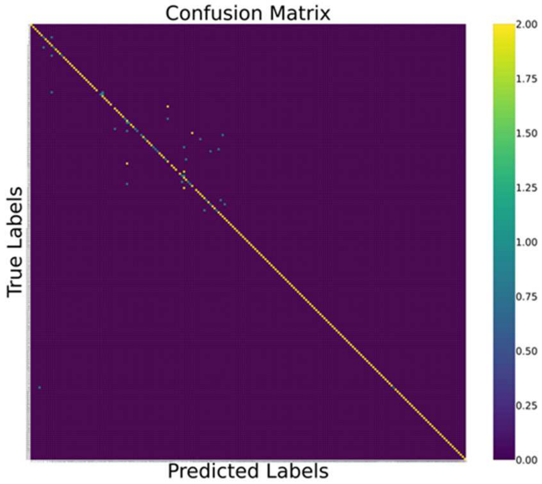

Fig. 4. Confusion matrix of the third model

It is noticed from this matrix that the majority of the features are correctly classified.

TABLE III. SIMULATION RESULTS FOR THE THIRD MODEL

	Value
Accuracy (%)	94
Precision	95
Recall	94
F1-score	94

Fig. 5 and Fig. 6 show the accuracy and loss for model 3, respectively. The accuracy graph illustrates the model's performance in terms of precision on the training and validation sets. The loss graph displays the decrease in the loss function, indicating how the model minimizes error during training.

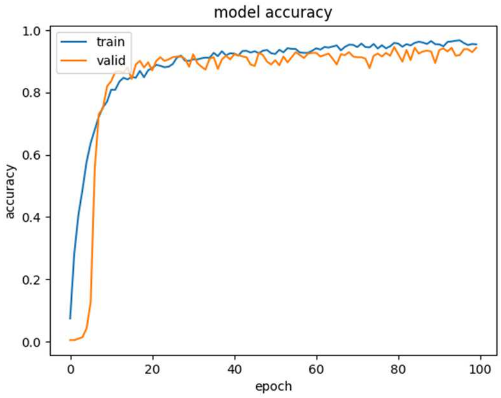

Fig. 5. The model's accuracy

Fig 6 shows the model's loss.

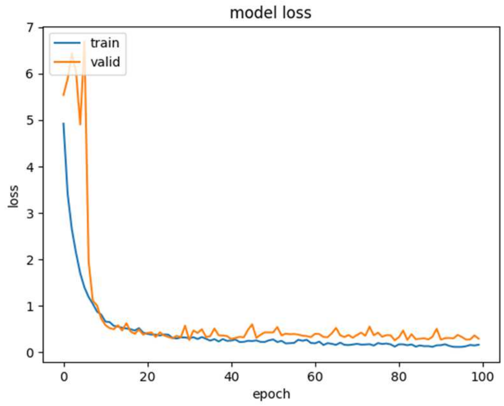

Fig. 6. The model's Loss

The model efficiency curve shows a steady increase in efficiency for the training and validation sets, indicating that the model learns effectively from the training data and generalizes well to the validation data. The model's loss curve reveals a rapid decrease in loss at the beginning, followed by a stabilization in both the training and validation sets, showing that the model effectively minimizes the loss function.

E. Face recognition using Raspberry pi

We are now approaching a crucial step which is the deployment of the Raspberry Pi model for real-time facial recognition. This phase marks a significant transition from the computing power of traditional computers to the compactness and limited resources of the Raspberry Pi device. Our goal is to create an autonomous and portable system, capable of recognizing faces in real time, thereby providing a practical solution for various applications in environments where access to conventional computers is limited or not available.

In this part, we will explore the essential steps to configure Raspberry Pi, transfer our trained CNN models, interface with a webcam to capture a real-time video stream, perform model inference for facial recognition and finally display the results on a screen or via a video output. We will also discuss the specific challenges and considerations to take into account when deploying deep learning models on an embedded device such as Raspberry Pi.

Once the model performs inference on the images captured by the webcam, we identify the detected faces and compare their features with those stored in our database. When a match is found, the matched person's name is displayed. This approach allows for instant identification of individuals and makes the facial recognition system more user-friendly and convenient to use.

F. Gui Attendance

After successfully accomplishing real-time facial recognition on Raspberry Pi, our next step is to develop a user-friendly graphical interface (GUI) with three key features.

In this paper, we propose to implement an automated attendance system. This interface will allow users to view the results of facial recognition, including the names of the identified people, as well as the date and time of their recognition.

Firstly, users can capture photos of themselves in different positions by specifying their ID and name for each set of photos using the 'Take Photos' button.

Then, the application allows us to train the facial recognition model with the data provided using the 'Train model' button. Finally, once the model is trained, users can test facial recognition in real time using the 'Test Recognition' button.

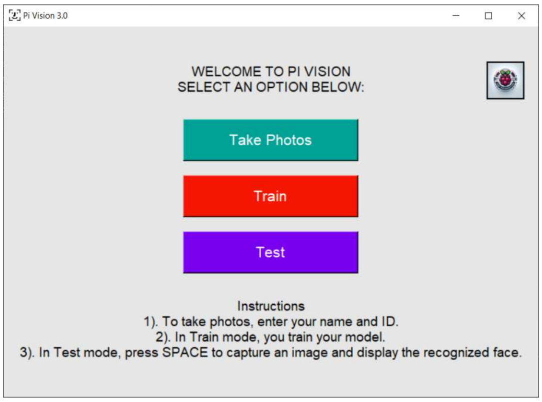

Fig. 7. GUI Interface

In addition, we will implement an automatic attendance recording system in the form of Excel files, thus facilitating the management of attendance data. These Excel files will contain relevant information such as the names of the people recognized and the corresponding dates. At the same time, we will integrate functionality to send these Excel files via email, providing a convenient solution for sharing attendance data with relevant parties.

Additionally, we will add the ability to control the Raspberry Pi remotely via a smartphone, providing additional flexibility in managing the facial recognition system. By combining an intuitive user interface, automated attendance system, email export functionality and remote control via smartphone, our system will become a versatile and effective tool for access management, security and monitoring presences in various environments.

Our facial recognition system is compatible with mobile devices via VNC Viewer, provided that the devices are connected in the same network. This integration allows users to access facial recognition features from their smartphone or tablet.

The system takes some time to run, we note the execution time in each of the environments in the table below:

TABLE IV. TIME PROCESSING COMPARISON

Deployment Environment	Facial Recognition Execution Time (s)
PC	13.24
Raspberry Pi 4	19.38

After implementing our facial recognition system on Raspberry Pi, the applications can expand to various industries, including universities, schools and businesses.

In universities and schools, our system greatly simplifies the management of student and student attendance. By automating the process of tracking class attendance, it reduces administrative burden for teachers and administrators while providing a reliable way to control access to secure on-campus facilities.

Likewise, in the professional world, our system streamlines the management of employee schedules and attendance. It allows precise monitoring of hours worked, thus facilitating payroll management and optimization of human resources. Additionally, our system also provides security benefits, quickly identifying authorized individuals and detecting potential intrusions. These applications in education and business demonstrate the significant impact our facial recognition system can have on the operational efficiency and security of organizations.

Although our facial recognition system on Raspberry Pi offers significant benefits in terms of attendance management and security, it also has some limitations to consider. First, facial recognition accuracy can be affected by environmental conditions such as lighting, camera quality, and viewing angles and shooting distance. Variations in these conditions can result in significant error rates and reduced system performance. Additionally, the reliability of facial recognition may be limited by physical changes such as hairstyles, glasses or accessories, which can alter a person's appearance and make recognition more difficult.

Our facial recognition system has a recognition range of 3 meters and a face detection capability of 3.5 meters, ensuring reliability and offering robust capabilities and performance for various applications. Fig 8 shows the result of recognition for a known and unknown individual.

-a- -b- -c-

Fig. 8. -a- Distant recognition -b- Unknown user -c- Multiple users

IV. CONCLUSION

We have developed a real-time facial recognition system on Raspberry Pi, using convolutional neural networks (CNN), with easy-to-use user interface integration and emailing functionality for attendance certificates. Despite the hardware constraints, significant progress has been made thanks to the use of CNN models which has allowed an accuracy rate of 94%. This system can be used in different fields such as security, presence system, smart home, assistance to the person etc. our system also provides security benefits, quickly identifying authorized individuals and detecting potential intrusions.

ACKNOWLEDGMENT

This work was supported by the Ministry of Higher Education and Scientific Research (MESRS) and the General Directorate of Scientific Research and Technological Development- DGRSDT (MESRS), Algerian Government.

REFERENCES

[1] Zamir M, Ali N, Naseem A, Ahmed Frasteen A, Zafar B, Assam M, Othman M, Attia E-A. Face Detection & Recognition from Images & Videos Based on CNN & Raspberry Pi. *Computation*. 2022; 10(9):148.

[2] G. S. Nagpal, G. Singh, J. Singh and N. Yadav, "Facial Detection and Recognition using OpenCV on Raspberry Pi Zero," *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Greater Noida, India, 2018, pp. 945-950.

[3] Pranav K B, Manikandan J,Design and Evaluation of a Real-Time Face Recognition System using Convolutional Neural Networks, Procedia Computer Science, Volume 171, 2020,Pages 1651-1659,ISSN 1877-0509.

[4] N. Mustakim, N. Hossain, M. M. Rahman, N. Islam, Z. H. Sayem and M. A. Z. Mamun, "Face Recognition System Based on Raspberry Pi Platform," *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, Dhaka, Bangladesh, 2019, pp. 1-4, doi: 10.1109/ICASERT.2019.8934485.

[5] K. Nakajima, V. Moshnyaga and K. Hashimoto, "A comparative study of conventional and CNN-based implementations of facial recognition on Raspberry-Pi," *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Herl'any, Slovakia, 2021, pp. 000217-000222.

[6] Zamir M, Ali N, Naseem A, Ahmed Frasteen A, Zafar B, Assam M, Othman M, Attia E-A. Face Detection & Recognition from Images & Videos Based on CNN & Raspberry Pi. *Computation*. 2022; 10(9):148.

[7] Ismail Mujahid Mukadam Computer Engineering / Ansari Mohammed Sajjad Shaikh Fuzail Shahnawaz Prof. Salim G. Shaikh Computer Engineering / "ATTENDANCE SYSTEM USING FACE RECOGNITION AND RASPBERRY PI – REVIEW", International Journal of IOT and Data Science (IJIDS) Volume 1, Issue 1, January-December 2023, pp. 1-6.

[8] Gupta I, Patil V, Kadam C, Dumbre S. Face detection and recognition using Raspberry Pi. In2016 IEEE international WIE conference on electrical and computer engineering (WIECONECE) 2016 Dec 19 (pp. 83-86). IEEE.

[9] Taye, M. M. (2023). Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions. Data Science and Artificial Intelligence, Philadelphia University, Amman 19392, Jordan.

[10] Kriegeskorte, N. (2015). Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing. Medical Research Council Cognition and Brain Sciences Unit, University of Cambridge, Cambridge CB2 7EF, United Kingdom.

[11] Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Florida Atlantic University*, 777 Glades Road, Boca Raton, FL, USA.

[12] Jogin, M., G D, D., Mohana, R. K., Meghana, R. K., Madhulika, M. S., & Apoorva, S. (2018). Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning. In *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology*.

[13] Sharma, N., Jain, V., & Mishra, A. (2018). An Analysis Of Convolutional Neural Networks For Image Classification. Paper presented at the International Conference on Computational Intelligence and Data Science (ICCID 2018), Amity University Uttar Pradesh, Noida, India.

[14] Salakhutdinov, R., Tenenbaum, J. B., & Torralba, A. (2013). Learning with Hierarchical-Deep Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[15] Pal, K. K., & Sudeep, K. S. (2016). Preprocessing for Image Classification by Convolutional Neural Networks. In Proceedings of the IEEE International Conference on Recent Trends in Electronics Information Communication Technology

[16] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012, pp. 4277–4280.

[17] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014

[18] Pranav, K. B., & Manikandan. (2020). Design and Evaluation of a Real-Time Face Recognition System using Convolutional Neural Networks. *Procedia Computer Science*, 171, 1651–1659.

[19] Zhou, Z., Wang, H., Shang, W., & Zhang, L. (Year). Image Segmentation Algorithms Based on Convolutional Neural Networks. *School of Computer Science, Communication University of China*, Beijing, China.

[20] Phung, V.H.; Rhee, E.J. A Deep Learning Approach for Classification of Cloud Image Patches on Small Datasets. J. Inf. Commun. Converg. Eng. 2018, 16, 173–178, doi:10.6109/jicce.2018.16.3.173