International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

System for Opening Cabinets and Equipment Containers Based on Facial Recognition

José Ignacio Vega Luna[1], José Francisco Cosme Aceves[1], Francisco Javier Sánchez Rangel[1], Gerardo Salgado Guzmán[1], Víctor Noé Tapia Vargas[1]
[1]Departmento de Electrónica-Área de Sistemas Digitales, Universidad Autónoma Metropolitana-Azcapotzalco, Ciudad de México, México

***Abstract*—*In certain types of facilities, such as data centers, laboratories and hospitals, cabinets or containers are used to install or store computing, telecommunications or measurement equipment that, due to their characteristics and utility, must be kept protected. In addition to using traditional opening mechanisms, such as padlocks, it is recommended to use at least one additional security method. One of the reliable biometric methods used in a wide variety of applications is facial recognition. This method that does not require contact with the person has become more important today due to the COVID-19 pandemic. Facial recognition is an ever-improving technology that is now based on artificial intelligence to provide greater certainty in authentication. The algorithms and models used in facial recognition require considerable computing power, however, hardware and software tools are now available to implement these algorithms in embedded devices. This paper presents a system for opening equipment cabinets and containers based on facial recognition. The system is based on an embedded ESP32-CAM module and the facial recognition software uses the MTMN model implemented through the ESP-IDF development framework. Images of the faces of users attempting to open the cabinet are sent to Firebase Cloud Storage for logging. The tests carried out showed that the system has an average response time of 156.78 ms and an average accuracy of 99.45 %.*

Keywords*—*Artificial intelligence, authentication, data centers, facial recognition, MTMN model, security method.

I. INTRODUCTION

Commonly, the cabinets and cages have a mechanical or electromechanical lock whose manipulation is carried out by the person responsible for access directly on site. With the mobility restrictions caused by the COVID-19 pandemic, the need arose to make the operation of some access systems and procedures more agile, safe and autonomous so that there is no contact between people. One of them is the opening of cabinets and equipment cages using facial recognition [1].

Facial recognition is the technology used to identify and recognize a person through the characteristics of the face. The methods and procedures used are based on photographic images, video or can be carried out in real time. Facial recognition is a biometric technology, similar to fingerprint, iris and voice recognition, used to allow access to a place, application or service. The goal of facial recognition is to automate the identification of people [2]. In general, facial recognition systems consist of the following stages [3], [4]:

- Image capture. Through a photographic or video camera, the image is captured in which the faces of the people are detected and located from the front or in profile, generating a file that in most cases is of the JPG type.
- Face analysis. Separate faces detected in the image are analysed to determine the geometry of the faces, including the shape of the cheekbones, lips, and mouth, as well as the distance from the eyes to the forehead or chin, and the shape of the ears. These features, or landmarks, are the basis for characterizing the face and are called special descriptors or face feature vectors.
- Digitization of the face. In this stage, the conversion of the reference points to digital values is carried out. One or more mathematical algorithms are used to obtain the unique digital identifier, or face ID, of the face or facial biometric pattern. This stage is also called mapping face features to numeric values, embedding, or encoding.
- Identification. The digital identifier of the face is searched and compared against an indexed set, or database, of known training faces. The objective is to determine the distance of the space of the features of the compared faces using the previously established threshold. Commonly, Euclidean distances, cosine distances or distances obtained based on a specific function are used to determine the similarity of the compared faces.

In practice, facial recognition systems carry out two actions: 1-They register the unique digital identifier of the face of a person authorized to access the service, installation or equipment, calling this action digital onboarding or enrollment, and 2-They recognize or authenticate a person who has previously been registered.

Currently, there are different facial recognition algorithms, both traditional, based on formulas and mathematical relationships, and those that use Artificial Intelligence (AI). Some of them are the following: the Viola–Jones object detection framework, the Eigenfaces or Principal Component Analysis (PCA) method, the Fisherfaces or Linear Discrimiant Analysis (LDA) classification method, the Gabor Wavelets method, the Hidden Markov model (HMM), the Active Shape Models (ASM) [5], [6] or a combination of the above [7].

Some of the algorithms mentioned above, and especially those that have emerged and been implemented in recent years, use AI techniques, machine learning, deep neural networks and computer vision [8]. These algorithms are trained using large databases of face images. The databases include faces with different quality characteristics, poses, traits and types that allow greater precision and reliability to authenticate people [9].

José Ignacio Vega Luna, José Francisco Cosme Aceves, Francisco Javier Sánchez Rangel, Gerardo Salgado Guzmán, and Víctor Noé Tapia Vargas, "System for Opening Cabinets and Equipment Containers Based on Facial Recognition," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 3, pp. 61-67, 2022.

Most of these algorithms are not based on complex mathematical formulas, like those used in traditional algorithms, to characterize a face and compare it with another. They depend on trained neural networks that use not only a reference point on the faces, but also a set of reference points expressed by means of pixels that indicate the features of the faces, or they combine one of the traditional algorithms with a method based on in AI [10]. Today there is a significant number of public face databases that can be used by the algorithm to be implemented [11], [12].

On the other hand, the growing use of mobile devices, social networks and the Internet of Things (IoT), has resulted in facial recognition being one of the most used biometric technologies for the identification and recognition of people, being used in a very wide spectrum. of human life and every day there is growing interest in making them more efficient and safer [13]. Today, they are used in the access and security of facilities and equipment, in the electronic market, in IoT applications, in health, in social networks, in banking and in surveillance systems, to name a few uses [14].

Due to the complexity, facial recognition algorithms require a large amount of computing power, however, recent advances in technology and the use of the IoT have led to the emergence of facial recognition application development platforms for development boards and SoCs. These platforms, some of them open source, are based on the use of powerful multi-core controllers, co-processors, communication interfaces and SDKs that allow the rapid and relatively simple implementation of facial recognition models and algorithms with neural networks and AI in embedded devices. With this, compact and efficient mobile facial recognition applications can be realized that work together with other technologies, or authentication devices, to provide a higher level of security as the application presented in this work.

The objective of this work was to make a system that allows the opening of a cabinet, or access door, where there are computer, measurement or clinical equipment of a laboratory, or data center, using facial recognition. The system registers on a microSD card the face IDs of the users authorized to open the equipment containers and sends to Firebase Cloud Storage the image of the face of the users who tried to open it, both the authorized ones, those that were successfully recognized, and the that were not recognized. The system was implemented using an ESP32-CAM embedded module as a base. Programming was carried out using the ESP-IDF development framework.

In certain applications, and especially in controlled environments, the use of a traditional facial detection and recognition method, sensitive to both the quality and quantity of the training images, is sufficient, since it provides acceptable results. However, the state-of-the-art of the works carried out in recent years in this field have focused mainly on addressing the challenge posed by Heterogeneous Face Recognition (HFR) [15]. HFR aims to solve the problems that arise in the variations in heterogeneous face images captured in uncontrolled conditions, such as changes of pose, illumination, background, occlusion, expression, video images [16] or images captured across different domains, the called visible to near-infrared images (NIR-VIS Face Recognition) [17] and facilitate the recognition tasks [18], [19]. HFR is focused on resolving discrepancies between images captured with different modalities, which is difficult to do with traditional algorithms based on feature descriptors, local binary patterns, scale-invariant feature transform and histogram of oriented gradients [20], [21]. Recent research proposes models based on: Generative Adversarial Networks (GAN) [22], Geometry Guided Pose-Invariant [23], reconstruction of the occluded part of the face [24], landmark curvature and vectorized landmark combining support vector machine (SVM) with a genetic algorithm (GA) [25], combinations of texture inpainting component and pose correction component [26], summation of 3D faces and cascaded regression in 2D and 3D shape spaces for face reconstruction [27], Face Augmentation Generative Adversarial Network (FA-GAN) to reduce the influence of imbalanced deformation attribute distributions using Graph Convolutional Networks (GCNs) [28], Adaptive Pose Alignment (APA) for Pose-Invariant Face Recognition [29], Dual-Agent Generative Adversarial Networks (DA-GANs) [30], Face Synthesis With Identity-Attribute Disentanglement [31], Domain discrepancy Elimination and Mean face Representation learning (DEMR) [32], Dual Face Alignment Learning (DFA L) [33] and partial face recognition approach, called dynamic feature matching (DFM), combining convolutional networks and sparse representation classification (SRC) to address partial face recognition problem regardless of various face sizes [34], among others. It is evident that the usefulness of the results derived from recent research is of great importance in these times of the COVID-19 pandemic, where most people wear a mask [35]. Most of the models proposed in recent years are based on convolutional neural network (CNN) whose purpose is to recognize faces regardless of the appearance resulting from human face changes over time, such as age [36] and other attributes such as gender, race and ethnicity [37], [38].

II. Methods

The realization of the system presented here was carried out by dividing it into two stages: the image capture and user interface hardware and the facial recognition software.

A. Image capture hardware and user interface

Fig. 1 shows the system hardware architecture. This circuit is installed in the upper middle part of the cabinet door, or container, which houses the computer, laboratory or measurement equipment. The two main elements of this stage are: the ESP32-CAM embedded module and the user interface.

The embedded module of the system hardware was the ESP32-CAM, which is based on the ESP-32S SoM. This module has the following features: dual core Tensilica Xtensa LX6 32-bit CPU, 520 KB SRAM, external 4 MB PSRAM, Wi-Fi 802.11b/g/n wireless interface with built-in antenna, IPEX connector for external antenna, UART, SPI, I2C, Bluetooth, and 4.2 BLE, ADC and DAC, socket for a MicroSD card, OmniVision Serial Camera Control Bus (SCCB) interface to connect an OV2640 or OV7670 video camera, and various GPIO (General Purpose Input/Output) terminals. It is compact

José Ignacio Vega Luna, José Francisco Cosme Aceves, Francisco Javier Sánchez Rangel, Gerardo Salgado Guzmán, and Víctor Noé Tapia Vargas, "System for Opening Cabinets and Equipment Containers Based on Facial Recognition," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 3, pp. 61-67, 2022.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

in size and low in price. In this work, an OV2640 camera was installed that has a 2 Megapixel CMOS sensor, which provides a UXGA photo resolution of 1622 x 1200 pixels, a video resolution of 1080p30, 720p60 and 640 x 480p90, and full frame. Images are sub-sampled, scaled, or 8-bit/10-bit in different formats such as JPG, BMP, and grayscale.

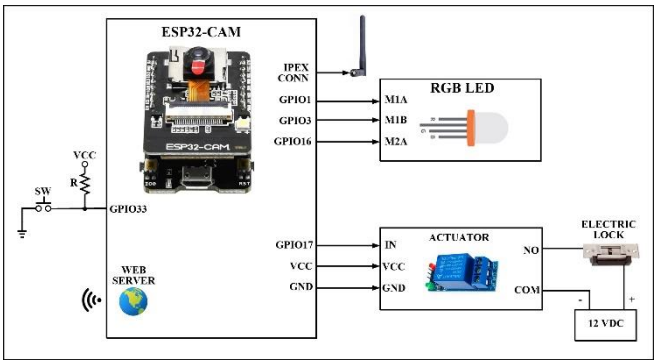

Fig. 1. System hardware architecture.

Additionally, the ESP32-CAM module has an IPEX connector to use an external Wi-Fi antenna to extend the range of wireless communication with a Wi-Fi access point. In this work a 2.4 GHz SMA to IPX Omnidirectional antenna was used which has a length of 8.8 cm.

The user interface is made up of an RGB led, a push-button and the electric lock actuator. The RGB led indicates the status of the system, similar to a traffic light. When the user is not authorized to open the cabinet, or the face was not recognized, the RGB led turns red. When the system is executing the facial recognition algorithm, the RGB led lights up yellow and when the system recognizes the user's face, the RGB led lights up green and the electric lock is activated. The push-button is used for the user to ask the system for authentication and recognition of the face and the opening of the cabinet. The RGB led terminals were connected to three GPIO terminals of the ESP32-CAM module configured as outputs, while the push-button was connected to a GPIO terminal configured as input. The electric strike is activated by a 12 VDC/10A/127 VAC relay module. The relay is activated via a GPIO pin configured as an output.

B. Facial recognition software

The system is operating in one of two states: 1-It waits for the user to press the push-button to request face recognition and the opening of the cabinet, or container, or 2-It waits for the request from the system administrator for registration, or enrollment, of a user. The actions necessary to implement the first state are executed in the main software program of the ESP32-CAM module, while a web server is executed in the background to implement the second state. Fig. 2 shows the flowchart used to perform system programming.

When starting execution, the main program performs the following tasks: it sets the environment variables, it configures the GPIOs used for the user interface, the OV2640 video camera and the microSD memory module. Next, it initializes the camera, the microSD card and the EEPROM memory, establishes the Wi-Fi connection with the access point, starts the web server in the background and enters a continuous cycle where it waits for the push-button to be activated to detect the face and try to recognize it.

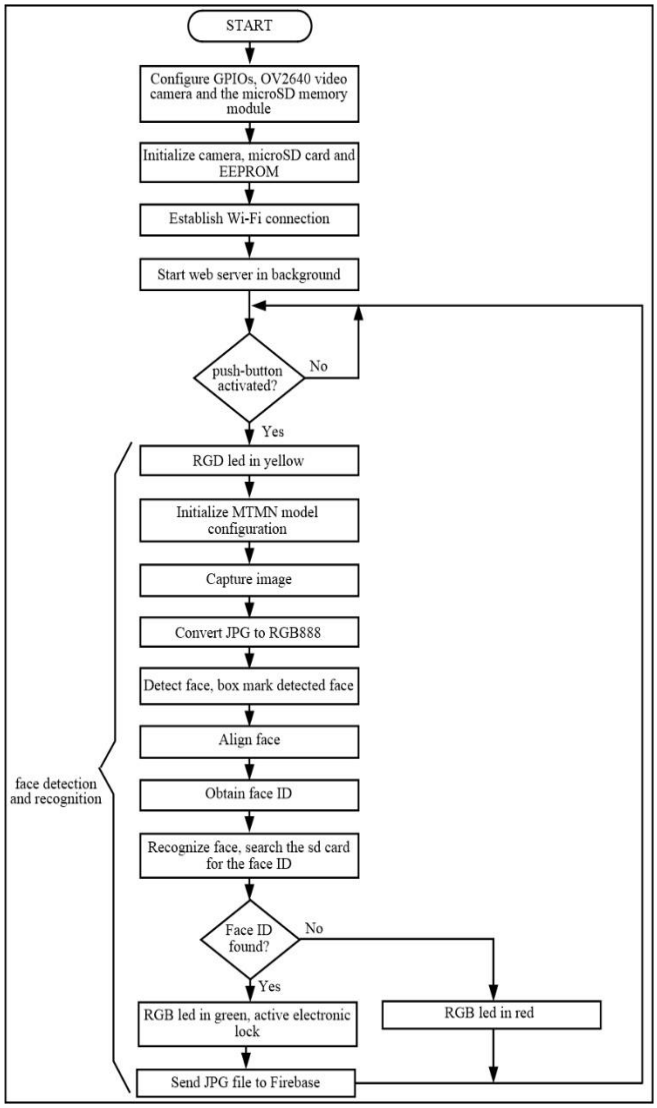

Fig. 2. Software flowchart.

To initialize the microSD card, the *SD_MMC.begin* function of the *FS.h* library was used and to establish the WiFi connection, the *WiFi.h* library was used. To initialize the EEPROM memory, the *EEPROM. begin* function of the *EEPROM.h* library was called.

The programming of the system was carried out using the V4.4 IoT ESP-IDF development framework. This open-source platform is used with the ESP32 SoCs. It works on Windows, Linux and macOS operating systems and provides a Software Development Kit (SDK) and a variety of APIs, software libraries and source code for application development. The majority of the components in ESP-IDF are available under the Apache 2.0 license. In this work, the ESP-IDF platform was installed using the Eclipse Integrated Development

José Ignacio Vega Luna, José Francisco Cosme Aceves, Francisco Javier Sánchez Rangel, Gerardo Salgado Guzmán, and Víctor Noé Tapia Vargas, "System for Opening Cabinets and Equipment Containers Based on Facial Recognition," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 3, pp. 61-67, 2022.

Environment (IDE) to create the source program, or script. The ESP-IDF platform integrates a Toolchain used to compile the program, create the project and upload the resulting code of the application presented here to the ESP32-CAM module, as indicated in Fig. 3. The ESP-IDF includes two function libraries, ESP-WHO and ESP-DL, to implement Neural Network (NN) Inference, Image Processing, Math Operations and high-performance Deep Learning Models. These libraries integrate APIs for detection and recognition of faces and face gesture, as well as for detection of movement, color, and hand poses.

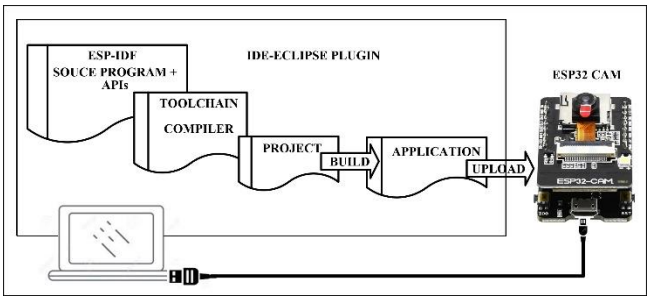

Fig. 3. ESP-IDF development framework.

The main reason why ESP-IDF was used to carry out the software for this work was because it provides, through ESP-WHO and ESP-DL, different advantages over other algorithms and face detection and recognition models that are explained. next. For face detection, ESP-WHO uses the MTMN (Multi-Task Memory Networks) model. This lightweight model is designed for embedded devices and its operation uses the mobile architecture called MobileNetV2. MobileNetV2 is based on Multi-task Cascaded Convolutional Networks (MTCCN). The size of the MTMN model is smaller than that of the MTCNN. MobileNetV2 represents the state-of-the-art of computer vision mobile models by decreasing the number of operations and memory needed, obtaining the same accuracy, or better, than other models. MobileNetV2 is made up of the next generation computer vision convolutional deep neural networks, which allows the implementation of real-time object classification, detection and alignment applications using personal mobile devices. MTMN uses pretrained neural networks based on more than a million images contained in the ImageNet database. The networks have learned feature representations of more than 1,000 object categories. MTMN's deep cascaded learning multi-task framework solves problems presented in an unconstrained environment caused by lighting, occlusions, and poses. This model has a cascaded structure composed of three stages of deep convolutional networks that predict face and landmark location in a coarse-to-fine way. The three stages are as follows: 1-Proposal Network (P-Net). This network proposes a candidate for bounding boxes. Sends the boxes to next network creating an image pyramid, in order to detect faces of all different sizes, 2-Refine Network (R-Net). Trace the bounding boxes in the P-Net and 3-Output Network (O-Net). Gets the final result, that is, the accurate bounding box, confidence coefficient and 5-point-landmark.

In this way, using the MTMN model and the function libraries *fd_forward.h* and *fr_forward.h* of ESP-IDF, the routine for face detection and recognition was developed, which is invoked from the main program after the user presses the button. push-button. This routine consists of the following tasks:

1-Turn on the RGD led in yellow.

2-Initializes the structure that stores the configuration of the MTMN model. This structure establishes, among other parameters, the minimum size of a detectable face, the image pyramid resizing times, the thresholds for P, R, and O Nets and the scale that controls the generated pyramids. In this work, the values of the parameters of this structure were the default configurations.

3-Captures the image of the user's face, in JPG format, using the *esp_camera_fb_get* function from the *esp_camera.h* function library.

4-Before performing the detection, a structure of type *dl-matrix3du-t* is defined, which stores the image to be processed. This structure is initialized by the *dl_matrix3du_alloc* function, which sets the following parameters used in the MTMN model: number of matrix3d, width of the matrix, which is the width of the image, height of the matrix, is equal to the height of the image and the number of channels, whose value is 3, since an RGB image is used.

5-The function to detect the face uses as inputs a pointer to the matrix *dl-matrix3du-t* which stores the bitmap of the image of the face in RGB888 format and the structure of the MTMN model configuration. In such a way that to convert the image from JPG to RGB888 format, the *fmt2rgb888* function was used.

6-Next, the *face_detec*t function is called to detect the face. This function returns the pointer to a structure that contains the points that delimit the box, or boxes, that mark the detected faces, as shown in Fig.4. The landmark coordinates of the faces are obtained.

Fig. 4. Face of the detected person.

7-Subsequently, the alignment of the face, or faces, of the image, marked in the box obtained previously, is carried out, invoking the *align_face* function. A structure is obtained that stores the information of the aligned face which will be used in the recognition process.

José Ignacio Vega Luna, José Francisco Cosme Aceves, Francisco Javier Sánchez Rangel, Gerardo Salgado Guzmán, and Víctor Noé Tapia Vargas, "System for Opening Cabinets and Equipment Containers Based on Facial Recognition," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 3, pp. 61-67, 2022.

8-The face ID of the aligned image is obtained using the *get_face_id* function.

9-Finally, an attempt is made to recognize the face by comparing the face ID obtained previously with those stored on the microSD card using the *recognizer.recognize* function. One of the results of this function is the similarity. If this result is equal to or greater than the reference value, set in the function settings, it is assumed that two Face IDs belong to the same user. The recognition is successful, the green RGB led lights up momentarily and the electronic lock is activated. If not, the RGB led lights up momentarily in red. In this step, the JPG file of the face image is sent to Firebase Cloud Storage to be stored in one of the directories named *successful* or *failed*, as the case may be.

To implement user enrollment, a routine was developed that runs in the background, which consists of the following tasks:

1-Using the functions of the *ESPAsyncWebServer.h* library, start the HTTP async server, which is waiting, listening on port 80, the connection of the system administrator.

2-Next, it waits for the user to press the push-button to ask the system to capture the image.

3-After the user presses the push-button, indicating that he is ready to capture the image, the RGB led turns red and tasks 4 to 8 of the detection and recognition routine explained above are executed.

4-Save, or register, in the microSD card, in a file called *faceX*, where *X* is the user number, the face ID of the detected user, using the functions *fs.open*, *file.write* and *file.close* of the library *SD_MMC.h*. Before using the microSD card, it was formatted with a Windows FAT32 type file system.

5-The number of registered users is updated in the EEPROM flash memory and the RGB led turns off. EEPROM memory was accessed using the *EEPROM.write* and *EEPROM.read* functions of the *EEPROM.h* library.

6- The JPG file of the face image is sent to Firebase Cloud Storage to be stored in the directory called enrollment.

Firebase is a platform in the cloud created by Google for the development of web and mobile applications in an easy and fast way. Firebase can be used on iOS, Android, web services and IoT sensors. It provides different functions including cloud storage, real time database, user authentication, crash reporting, application test lab and remote config, cloud messaging and hosting. Firebase Cloud Storage was designed for portability. It is aimed at creating applications that need a user-generated object storage service, such as photos, audio, and video. It has a set of APIs that can be used from the application program to access the files stored on the platform. In the development of this work, the following tasks were carried out in the Firebase Cloud Storage Console before using the platform from the ESP32-CAM module software: 1-A project containing the service configuration was created, 2-It was established the Authentication Method, in this case Email/Password is used as the sign-in method, 3-The user and password were added, 4-The Storage Bucket was created, which returns the storage bucket ID and 5-The Project API Key is obtained.

In the software section of the ESP32-CAM module that uploads to Firebase Cloud Storage the captured images of users trying to access the system, the Firebase ESP client library, *Firebase_ESP_Client.h*, was used, which has the API key, the storage arguments. bucket ID project and the path where the JPG files are stored.

III. Results and Discussion

Fig. 5 shows a photograph of the developed system used in the cabinet door where some servers are installed.

In the programming carried out, data structures are used that establish the values of variables and parameters used by the functions invoked to initialize and configure the hardware and software components of the system.

Fig. 5. System developed and installed.

In most of the structures, the default value of the variables was used. One of the most important functions, which is essential to achieve acceptable results in face recognition, is *face_detect*. This function has as inputs the structure that contains the image to be processed to detect faces and the structure that stores the parameters to be used in the MTMN model. In these structures, two important variables are established that impact the behavior of the system. One is the MTMN model used, and the other is the value of the quantization type. The MTMN model can be: MTMN lite in quantization (default), MTMN lite in float or MTMN heavy in quantization and the value of the second variable can be: 8-bit quantization (default) or 16-bit quantization. Considering the above, the value of these variables was modified to carry out two sets of tests.

The first group of tests aimed to determine the appropriate value of these variables to use in the system and obtain the best performance. These tests were carried out trying to recognize the face of 1,000 users, of which a number were registered, authorized to open the container, and the rest were not authorized. The tests consisted of using each of the three MTMN models with each of the two quantization types. That is, six tests were carried out with the 1,000 users and both the average response time and the percentage of accuracy of the system when trying to recognize the face were determined. The results of these tests are indicated in Table I. It can be seen that the 8-bit quantization type has lower accuracy than the 16-bit

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

quantization type but is faster, the response time is lower, shorter latency. Given the above, it was chosen to use the MTMN lite in quantization model and the 8-bit quantization type because the shortest response time and acceptable accuracy are obtained.

TABLE I. System performance and accuracy.

	Performance, response time (ms)- Accuracy %	
	8-bit quantization (default)	**16-bit quantization**
MTMN lite in quantization (default)	156.78-99.45	186.65-99.65
MTMN lite in float	185.78-99.65	190.78-99.65
MTMN heavy in quantization	209.87-99.15	217.56-98.35

The second set of tests aimed to determine the reliability of the system through the percentage of accuracy achieved. The tests consisted of establishing the reference value of the similarity parameter with different values in the face recognition function. The results of these tests are shown in the graph of Fig. 6, from which it can be seen that for established similarity values equal to or greater than 0.9985, the percentage of accuracy remains almost constant, 99.45 %, and for values less than 0.9985 it decreases significantly.

It was possible to develop the system hardware using another embedded module, controller or SoC with more resources. However, the application carried out requires the system to be as compact as possible since it is placed on a door, which implies that the installation process must not be intrusive and must not affect the operation of the door by adding components, or devices, considerable size and weight. Additionally, one of the objectives of the application was to obtain a low-cost and easy-to-operate system.

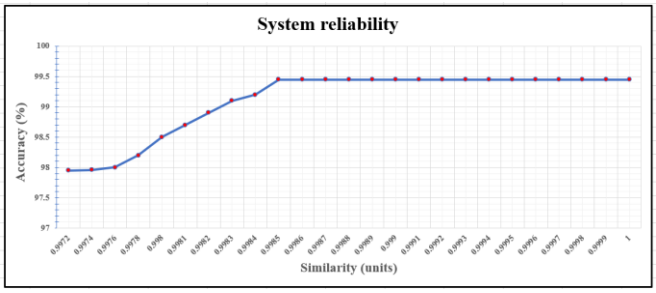

Fig. 6. System reliability.

Finally, the ESP32-CAM module supports microSD cards up to 4 GB. However, a 64 GB microSD card was used in this work to store the users' face IDs and there were no problems accessing the card.

IV. Conclusion

A system for opening cabinets, or containers, of computer, laboratory or measurement equipment was built using facial recognition as a security method and an ESP32-CAM embedded module that integrates an OV2640 video camera. If the system recognizes the face of the user requesting the opening of the cabinet, it activates its electric lock. JPG files of the faces of people who request to open, both successfully and unsuccessfully, are sent to the Firebase Cloud Storage platform for logging. The system administrator can enroll users through a web server that is running, in the background, in the ESP32-CAM module software. In this process, the software determines the user's face ID and stores it on the ESP32-CAM's microSD card. When the user requests the opening of the cabinet, the facial recognition algorithm obtains the user's face ID and compares it with the face IDs stored in the microSD to obtain the similarity between both and make the decision on the activation of the electric lock.

The model, or algorithm, of facial recognition used was the MTMN (Multi-Task Memory Networks), which is based on three convolutional neural networks. The system software, including the MTMN model, was implemented using the ESP-IDF Internet development platform and open-source function libraries. The advantage of using this platform is that it integrates AI, computer vision and deep learning libraries with neural networks trained on public databases containing a very large number of face images. This had the benefit of reducing the development time of the application, obtaining a secure and reliable system with an average accuracy of 99.45%.

Currently, the system only allows the system administrator to enroll new users. However, work is being done to incorporate the option to remove registered users and integrate, both the face ID stored on the microSD card and the JPG file stored on Firebase, the name of the users. These actions require integrating the corresponding code into the system software without modifying the hardware.

References

[1] H. Dong, A. Munir, H. Tout and Y. Ganjali, "Next-Generation Data Center Network Enabled by Machine Learning: Review, Challenges, and Opportunities", *IEEE Access*, vol. 9, pp. 136459-136475, 2021.

[2] M. O. Oloyede, G. P. Hancke and H. C. Myburgh, "Improving Face Recognition Systems Using a New Image Enhancement Technique, Hybrid Features and the Convolutional Neural Network", *IEEE Access*, vol. 6, pp. 75181-75191, 2018.

[3] Q. Feng, C. Yuan, J. -S. Pan, J. -F. Yang and Y. -T. Chou, "Superimposed Sparse Parameter Classifiers for Face Recognition", *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 378-390, 2017.

[4] Y. Zhong, J. Chen and B. Huang, "Toward End-to-End Face Recognition Through Alignment Learning", *IEEE Signal Processing Letters*, vol. 24, no. 8, pp. 1213-1217, 2017.

[5] Y. Ding, Q. Zhao, B. Li and X. Yuan, "Facial Expression Recognition From Image Sequence Based on LBP and Taylor Expansion", *IEEE Access*, vol. 5, pp. 19409-19419, 2017.

[6] J. Chen, J. Chen, Z. Wang, C. Liang and C.-W. Lin, "Identity-Aware Face Super-Resolution for Low-Resolution Face Recognition", *IEEE Signal Processing Letters*, vol. 27, pp. 645-649, 2020.

[7] J. Kong, M. Chen, M. Jiang, J. Sun and J. Hou, "Face Recognition Based on CSGF(2D)2PCANet", *IEEE Access*, vol. 6, pp. 45153-45165, 2018.

[8] B. Yang, J. Cao, R. Ni and Y. Zhang, "Facial Expression Recognition Using Weighted Mixture Deep Neural Network Based on Double-Channel Facial Images", *IEEE Access*, vol. 6, pp. 4630-4640, 2018.

[9] G. Lou and H. Shi, "Face image recognition based on convolutional neural network", *China Communications*, vol. 17, no. 2, pp. 117-124, 2020.

[10] J. Y. Choi and B. Lee, "Ensemble of Deep Convolutional Neural Networks With Gabor Face Representations for Face Recognition", *IEEE Transactions on Image Processing*, vol. 29, pp. 3270-3281, 2020.

[11] J. Muhammad, Y. Wang, C. Wang, K. Zhang and Z. Sun, "CASIA-Face-Africa: A Large-Scale African Face Image Database", *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3634-3646, 2021.

[12] H. Yang and X. Han, "Face Recognition Attendance System Based on Real-Time Video Processing", *IEEE Access*, vol. 8, pp. 159143-159150, 2020.
[13] D. Liu, N. Bellotto and S. Yue, "Deep Spiking Neural Network for Video-Based Disguise Face Recognition Based on Dynamic Facial Movements", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 1843-1855, 2020.
[14] C. Ding and D. Tao, "Trunk-Branch Ensemble Convolutional Neural Networks for Video-Based Face Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 1002-1014, 2018.
[15] D. Liu, X. Gao, N. Wang, J. Li and C. Peng, "Coupled Attribute Learning for Heterogeneous Face Recognition", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4699-4712, 2020.
[16] F. Mokhayeri, E. Granger and G. Bilodeau, "Domain-Specific Face Synthesis for Video Face Recognition from a Single Sample Per Person", *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 757-772, 2019.
[17] R. He, X. Wu, Z. Sun and T. Tan, "Wasserstein CNN: Learning Invariant Features for NIR-VIS Face Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1761-1773, 1 July 2019.
[18] K. Zhang, Y. Huang, Y. Du and L. Wang, "Facial Expression Recognition Based on Deep Evolutional Spatial-Temporal Networks", *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4193-4203, 2017.
[19] X. Geng, Z. H. Zhou and K. Smith-Miles, "Individual Stable Space: An Approach to Face Recognition Under Uncontrolled Conditions" *IEEE Transactions on Neural Networks*, vol. 19, no. 8, pp. 1354-1368, 2008.
[20] C. Fu, X. Wu, Y. Hu, H. Huang and R. He, "DVG-Face: Dual Variational Generation for Heterogeneous Face Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 2938-2952, 2022.
[21] D. Gong, Z. Li, W. Huang, X. Li and D. Tao, "Heterogeneous Face Recognition: A Common Encoding Feature Discriminant Approach", *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2079-2089, 2017.
[22] X. Zhang, F. Zhang and C. Xu, "Joint Expression Synthesis and Representation Learning for Facial Expression Recognition", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1681-1695, 2022.
[23] F. Zhang, T. Zhang, Q. Mao and C. Xu, "Geometry Guided Pose-Invariant Facial Expression Recognition", *IEEE Transactions on Image Processing*, vol. 29, pp. 4445-4460, 2020.
[24] D. Poux, B. Allaert, N. Ihaddadene, I. M. Bilasco, C. Djeraba and M. Bennamoun, "Dynamic Facial Expression Recognition Under Partial Occlusion With Optical Flow Reconstruction", *IEEE Transactions on Image Processing*, vol. 31, pp. 446-457, 2022.
[25] X. Liu, X. Cheng and K. Lee, "GA-SVM-Based Facial Emotion Recognition Using Facial Geometric Features", *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11532-11542, 2021.
[26] R. He, J. Cao, L. Song, Z. Sun and T. Tan, "Adversarial Cross-Spectral Face Completion for NIR-VIS Face Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 5, pp. 1025-1037, 2020.
[27] F. Liu, Q. Zhao, X. Liu and D. Zeng, "Joint Face Alignment and 3D Face Reconstruction with Application to Face Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 3, pp. 664-678, 2020.
[28] M. Luo, J. Cao, X. Ma, X. Zhang and R. He, "FA-GAN: Face Augmentation GAN for Deformation-Invariant Face Recognition", *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2341-2355, 2021.
[29] Z. An, W. Deng, J. Hu, Y. Zhong and Y. Zhao, "APA: Adaptive Pose Alignment for Pose-Invariant Face Recognition", *IEEE Access*, vol. 7, pp. 14653-14670, 2019.
[30] J. Zhao, L. Xiong, J. Li, J. Xing, S. Yan and J. Feng, "3D-Aided Dual-Agent GANs for Unconstrained Face Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 10, pp. 2380-2394, 2019.t
[31] Z. Yang, J. Liang, C. Fu, M. Luo and X. -Y. Zhang, "Heterogeneous Face Recognition via Face Synthesis With Identity-Attribute Disentanglement", *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1344-1358, 2022.
[32] W. Hu and H. Hu, "Domain Discrepancy Elimination and Mean Face Representation Learning for NIR-VIS Face Recognition", *IEEE Signal Processing Letters*, vol. 28, pp. 2068-2072, 2021.
[33] W. Hu, W. Yan and H. Hu, "Dual Face Alignment Learning Network for NIR-VIS Face Recognition", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2411-2424, 2022.
[34] L. He, H. Li, Q. Zhang and Z. Sun, "Dynamic Feature Matching for Partial Face Recognition", *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 791-802, 2019.
[35] Y. Martínez-Díaz, H. Méndez-Vázquez, L. S. Luevano, M. Nicolás-Díaz, L. Chang and M. González-Mendoza, "Towards Accurate and Lightweight Masked Face Recognition: An Experimental Evaluation", *IEEE Access*, vol. 10, pp. 7341-7353, 2022.
[36] J. Zhao, S. Yan and J. Feng, "Towards Age-Invariant Face Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 474-487, 2022.
[37] L. Jiang, J. Zhang and B. Deng, "Robust RGB-D Face Recognition Using Attribute-Aware Loss", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2552-2566, 2020.
[38] L. Best-Rowden and A. K. Jain, "Longitudinal Study of Automatic Face Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 148-162, 2018.